Bivariate pseudospectral collocation algorithms for nonlinear partial differential equations

by

Vusi Mpendulo Magagula

Submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Applied Mathematics

School of Mathematics, Statistics and Computer Science

University of KwaZulu-Natal

November 2016

This thesis is dedicated to the Magagula Family, especially my late father Mr. Mhhomzane Nathaniel Magagula.

DECLARATION 1 - STUDENT AND SUPERVISORS

The work described in this thesis was carried out under the supervision of Prof. S.S. Motsa and Prof. P. Sibanda in the School of Mathematics, Statistics and Computer Science, College of Agriculture, Engineering and Science, University of KwaZulu-Natal, Pietermaritzburg, from January 2014 to October 2016.

I hereby declare that no portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institution of learning. The thesis is my original work except where due reference and credit is given.

Signature: ....

Magagula Vusi Mpendulo Date

Signature: ....

Prof. S.S. Motsa Date

Signature: ....

Prof. P. Sibanda Date

DECLARATION 2 - PLAGIARISM

I, **Vusi Mpendulo Magagula** declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:
   (a) Their words have been re-written but the general information attributed to them has been referenced.
   (b) Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.
5. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signature: ....

Magagula Vusi Mpendulo Date

DECLARATION 3 - PUBLICATIONS

Details of contribution to publications that form part or include research presented in this thesis (include publications in preparation, submitted, in press and published and give details of the contributions of each author to the experimental work and writing of each publication). All these papers were written by the student under the supervision of Prof. S.S. Motsa and Prof. P. Sibanda.

1. Motsa, S. S., Magagula, V. M., & Sibanda, P. (2014). A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations. The Scientific World Journal, Article ID 581987, 13 pages. doi:10.1155/2014/581987

2. Magagula, V. M., Motsa, S. S., Sibanda, P., & Dlamini, P. G. (2016). On a bivariate spectral relaxation method for unsteady magneto-hydrodynamic flow in porous media. SpringerPlus, 5(1), 1.

3. Magagula, V. M., Motsa, S. S., & Sibanda, P., (2016). A multi-domain bivariate pseudospectral method for evolution equations. International Journal of Computational Methods, 27 pages, doi: 10.1142/S0219876217500414

Signature: ....

Magagula Vusi Mpendulo Date

Acknowledgements

This research project could not have been accomplished without the help and support of my supervisors, my family, and friends. In particular, I would like to acknowledge the contributions of the following outstanding people for their support, motivation, and prayers throughout this journey.

Foremost, giving thanks to my creator the Lord God almighty for providing me with life, strength and many abilities to complete this research project.

To the research project advisors Professors Sandile S. Motsa and Precious Sibanda for all the invaluable guidance and assistance that they have provided me with to make sure that this thesis was a great success. I regard them as my role models, excellent researchers, and inspirational academics. In addition I would like to thank all members of the mathematics faculty who provided timely and invaluable feedback during presentations at the University of KwaZulu-Natal, Pietermaritzburg. Professor François Blanchette, thanks for guiding me through my Masters in Applied Mathematics while I was a student at the University of California, Merced. Professor Martin J. Mohlenkamp, thanks for the research experience and opportunities you provided me with while a postgraduate student at Ohio University. Prof. S.S. Motsa, thank you for introducing and exposing me to research while I was an undergraduate student. Dr P.G. Dlamini many thanks for your valuable advice and suggestions when I experienced difficulties on this journey.

To my lovely parents who have provided me with invaluable support throughout my education journey. You were always there for me, believing in me and giving me countless opportunities to develop as an aspiring Mathematician. You enriched this journey I would not have made it thus far without you. My lovely sister's Nomkhosi, Andile, Tenele and Bethusile thank you for encouraging me and I must say your presence in my life contributed immensely on this journey. Thank you so kindly bo Mtfombeni for understanding my long stay in the academic

field. Dr Reuben Dlamini, Dr Gcina Mavimbela and Dr Thembinkosi Mkhatshwa, thank you for being there brothers and providing constant support and great encouragement. You have been inspirational and supportive at all times. Babe Zombodze R. Magagula, thanks for all your invaluable support and advice. Mr. S.S. Bulunga, thanks for being a role model and for providing me with foundational mathematics skills.

Finally, my sincere appreciation to all my friends and colleagues for making my stay at the University of KwaZulu-Natal, Pietermaritzburg an unforgettable one, especially the members of the Numerical Analysis, Mathematical Biology and Fluid Dynamics Journal Clubs. Dr Z.G. Makukula, thanks mkhaya for orienting me to Pietermaritzburg and the constructive critique in my initial stages of the project. Sicelo Goqo , thanks for being there when I needed someone to vent my research frustrations to and for the nerdy conversations. Dr R. Mpika-Massouku, thanks for being a good and understanding office mate. To the administrative staff and technicians, Bev, Christel, Jabulile, Jessica, Faith (may her soul rest in peace), Sandile and Shaun, thanks for your professionalism. Bongiwe P. Hlanze, aka Queen B., thanks for the emotional and endless support in the last lap of this research project. Thanks Sheelagh Halstead for your editing services. You raised good points which improved the outlook and presentation of the thesis.

The financial support of the DST-NRF Centre of Excellence in Mathematical and Statistical Sciences (CoE-MaSS) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the CoE.

The financial support of Prof. S.S. Motsa, Prof. P. Sibanda, the school of Mathematics, Statistics and Computer Sciences (MSCS), College of Agriculture, Engineering and Science and the University of KwaZulu-Natal (UKZN) towards this research is also hereby acknowledged. Opinions expressed and conclusions arrived at, are those solely of the author and are not necessarily to be attributed to the above mentioned persons and entities.

Abstract

We present seven modified pseudospectral collocation methods for solving nonlinear evolution differential equations, and systems of nonlinear partial differential equations. They converge in a fraction of a second, use few grid points and are accurate. These numerical methods are implemented using spectral collocation, independently both in space and time. They are presented in general form, are then used to solve nonlinear evolution differential equations and systems of nonlinear partial differential equations arising from fluid dynamics. They are compared with the bivariate pseudospectral methods and are validated to show accuracy.

The bivariate spectral quasilinearization method (BSQLM), for nonlinear evolution equations is introduced in a general form for pth order nonlinear evolution equation. Applicability, and accuracy of the method is confirmed by solving one dimensional nonlinear partial differential equations. The method is extended to a system of non-similar partial differential equations that model magnetohydrodynamic forced convection flow adjacent to a non-isothermal wedge. The bivariate spectral relaxation method (BSRM) is presented for systems of four nonlinear partial differential equations modeling an unsteady three dimensional magnetohydrodynamic flow, and mass transfer, in a porous media. Compared with previously published results, the BSRM is computationally efficient and converges quickly with few grid points. The performance of three bivariate pseudospectral methods is analyzed. The generalized bivariate spectral quasilinearization method (BSQLM) and bivariate spectral local linearization method (BSLLM) for systems of n nonlinear partial differential equations are presented for the first time in general form and are compared with the BSRM. For each method, convergence and residual graphs, tables of convergence errors and computational time are presented. A modified bivariate spectral quasilinearization method is then presented, which uses Legendre-Gauss-Lobbatto grid points instead of the Lagrange-Gauss-Lobbatto grid points. This approach improves the accuracy of

the bivariate spectral quasilinearization method. It is tested on nonlinear partial differential equations.

The multi-domain bivariate spectral quasilinearization method (MD-BSQLM) for nonlinear evolution equations is introduced in a general form for solving pth order nonlinear evolution equations over a large time interval. Its applicability, and reliability is confirmed by solving one dimensional nonlinear partial differential equations over a large time domain. The pseudospectral method is extended to a system of nonlinear coupled non-similar boundary layer partial differential equations over a large independent variable interval. It is developed for systems of n coupled non-similar boundary layer partial differential equations. Lastly, a multi-domain bivariate spectral local linearization method (MD-BSLLM), is developed for solving system of n coupled non-similar partial differential equations whose solutions have boundary layers. As before, the MD-BSLLM is tested for nonlinear coupled non-similar boundary layer partial differential equations over a large independent variable interval.

Publications

The work in this thesis has been published in the following journals. The publications are attached at the end of this thesis.

1. Motsa, S. S., Magagula, V. M., & Sibanda, P. (2014). A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations. The Scientific World Journal, Article ID 581987, 13 pages. doi:10.1155/2014/581987

2. Magagula, V. M., Motsa, S. S., Sibanda, P., & Dlamini, P. G. (2016). On a bivariate spectral relaxation method for unsteady magneto-hydrodynamic flow in porous media. SpringerPlus, 5(1), 1.

3. Magagula, V. M., Motsa, S. S., & Sibanda, P., (2016). A multi-domain bivariate pseudospectral method for evolution equations. International Journal of Computational Methods, 27 pages, doi: 10.1142/S0219876217500414

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Literature Review

Naturally occurring phenomena and their respective dynamics can be captured accurately using nonlinear partial differential equations. Nonlinear evolution equations reveal some notable features of physical systems. For instance, the Burgers equation has been used as a mathematical model of turbulence [1], the KdV-Burgers equation is applicable in quantum field theory, plasma physics and solid-state physics [2], the Burgers–Huxley equation describes the interaction between reaction mechanisms, convection effects and diffusion transports [3] while the FitzHugh-Nagumo equation models the transmission of nerve impulses and has also been used in population genetics and electrical circuit theory [4]. In fluid mechanics, boundary layer problems are represented using systems of nonlinear partial differential equations. In as much as we model these naturally occurring physical systems using complex nonlinear partial differential equations, it is often difficult or impossible to obtain exact solutions. Some of the most interesting and important features of naturally occurring systems can only be studied with methods designed to tackle their nonlinearity. Nevertheless, despite these difficulties, researchers have developed analytical techniques for finding solutions of such nonlinear complex partial differential equations. These methods include the homotopy analysis method [5–10], homotopy perturbation method [11–16] and variational iteration method [17–21]. These methods all use a series approach, and hence their accuracy is determined by the number of terms used in the series. For complex systems of partial differential equations, many terms would be required, which

then leads to complicated expressions which may not be simplified by mathematics software currently available. Thus, such analytical methods are not ideal for solving nonlinear partial differential equations.

Numerical methods have, instead, been developed to overcome the challenges encountered in analytic methods. Finite difference based methods, such as the Keller-box method [22–25] have been developed for solving boundary layer equations. However, finite difference based methods require a fine grid to converge to an exact solution. A fine grid implies that many values must be calculated for the many grid points. Therefore, for accurate results, this numerical method demands considerable computational time, which becomes worse with large systems of equations. Furthermore, in some instances, finite difference methods may not converge to the exact solution, despite a fine grid. In a bid to decrease computational time and hasten convergence, spectral methods have been combined with finite difference based methods. In Section 1.2, we discuss the advantages and disadvantages of numerical methods. Solutions to overcome disadvantages of some of the numerical methods are being discussed. Aims and Objectives of this project are presented Section 1.3. The outline of this thesis is presented in Section 1.4.

1.2 Existing Numerical Methods

In this section, we discuss some numerical methods that have been used to solve nonlinear partial differential equations. We discuss their properties, advantages and disadvantages, together with some solutions to overcome the disadvantages.

1.2.1 Finite Differences Methods

Finite differences are the simplest of numerical methods and are a popular approach for obtaining numerical solutions of differential equations. They are discretization methods for solving differential equations by approximating them with difference equations, in which finite differences approximate the derivatives. However, their conditional stability depends on the physical problem being modeled. Finite differences are suitable for regular grids. Boundary conditions are easier to implement for regular grids when using finite differences [26].

Parabolic partial differential equations may be approximated by explicit or implicit finite differences [26, 27]. The explicit finite difference method uses forward difference in time and central difference in space. For this method, one needs to know the values at the current time interval so as to determine the values at the next time interval, and hence this is termed an explicit method. In this case, convergence and stability depend on the ratio of the time and space steps. By contrast, the implicit method solves an equation using both the current state of the system and a later one. Using this method is tedious and intensive, since it requires solving a system of equations.

Finite difference methods depend on approximating a function by a local polynomial interpolant. The derivatives of the function are then approximated by differentiating the local polynomial. By local, we mean the use of finely spaced grid points to approximate the function or its derivative at a point. Low degree local polynomials require a fine grid in order to accurately resolve a function. Thus, we need numerical methods that would allow coarser grids and fewer computational resources. Spectral methods are such methods because they use all available function values to construct the required approximations. A brief review of spectral methods follows next.

1.2.2 Spectral Methods

Spectral methods are numerical techniques that can achieve high accuracy results with comparatively few grid points because they use all available function values to construct the required approximations for differential equations. They are thus called global methods [28–30]. The two main components of spectral methods are the approximating functions and weight functions. The approximating functions are a linear combination of suitable basis functions, which provide an approximate representation of the solution. Then the weight functions ensure that boundary conditions of the differential equations are satisfied.

The most commonly used basis functions are Legendre [31, 32], Lagrange [33, 34], Laguerre [35, 36] and Chebyshev [29, 30] polynomials. With equi-spaced points approximations can diverge near the endpoints, which is known as the Runge phenomenon for non-periodic functions. Therefore, when interpolating smooth functions, Chebyshev based grid points are the preferred

choice because they provide approximations with nearly uniform accuracy over the closed interval $[-1,1]$.

In the general form of spectral methods, we seek a solution of the form

$$f(x) \approx \sum_{i=0}^{N} f(x_i)\phi_i(x) \tag{1.1}$$

where $\phi_i(x)$ is an interpolating polynomial, and $f(x_i)$ is the value of the function $f(x)$ evaluated at the grid points x_i. There are various types of Chebyshev grid points. For instance, three given by Hesthaven [28] and Boyd [29] are shown as:

$$x_i = \cos\left(\frac{(2i+1)\pi}{(2N+2)}\right), \quad \text{Chebyshev-Gauss} \tag{1.2}$$

$$x_i = \cos\left(\frac{(2i\pi)}{(2N+1)}\right), \quad \text{Chebyshev-Gauss-Radau} \tag{1.3}$$

$$x_i = \cos\left(\frac{\pi i}{N}\right), \quad \text{Chebyshev-Gauss-Lobatto} \tag{1.4}$$

In this thesis, we will use Lagrange polynomials with Chebyshev-Gauss-Lobatto and Legendre-Gauss-Lobatto grid points. The Legendre-Gauss grid points are zeros of the $(N+1)$th Legendre polynomial, Legendre-Gauss-Radau grid points are the zeros of the sume of the Nth and $(N+1)$th Legendre polynomial. The end points of the Legendre-Gauss-Lobatto grid points are $-1,1$ and the interior points are the zeros of the first derivative of the Nth Legendre polynomial. The characteristic Lagrange cardinal polynomial denoted by $L_i(x)$ is given by

$$L_i(x) = \prod_{\substack{i=0 \\ i\neq k}}^{N} \frac{x - x_k}{x_i - x_k}, \tag{1.5}$$

where

$$L_i(x_k) = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k \end{cases} \tag{1.6}$$

The first derivative of the Legendre polynomial is given by

$$P'_{N_x}(x) = \frac{N(N+1)}{(2N+1)}\left[\frac{P_{N-1}(x) - P_{N+1}(x)}{1-x^2}\right]. \tag{1.7}$$

Partial differential equations have been solved using spectral collocation (in space) and finite differences (time). The partial differential equation is first expressed as an ordinary differential equation, which is then solved using the Runge-Kutta method [37–40]. However, because finite difference methods are known to converge more slowly than spectral methods, combining the two would compromise the rapid convergence of spectral methods. Accordingly, in this thesis, we aim to develop spectral methods that can be used to solve nonlinear partial differential equations in both space and time, yet will still, converge quickly achieve accurate results with few grid points. In the next subsection, we first review the existing spectral methods.

1.2.3 A Combination of Finite Differences and Spectral Methods

Combined finite differences and spectral methods that are used to approximate solutions of nonlinear differential equations. The methods utilized the spectral method to discretize derivatives in space and finite differences to discretize in time. These methods include spectral quasilinearization method (SQLM), spectral relaxation method (SRM) and spectral local linearisation method (SLLM).

Applying the spectral method in space improved the accuracy, convergence and computational time of the finite differences methods. However, the accuracy, convergence and computational time of these methods are compromised. In the next subsection, we introduce new methods that will overcome these disadvantages.

1.3 Modified Numerical Methods

1.3.1 Bivariate Spectral Methods

In this subsection, we present the spectral methods that are used to approximate solutions of nonlinear differential equations in both space and time. The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u(x,t) \approx \sum_{i=0}^{N_x} \sum_{j=0}^{N_t} u(x_i,t_j) L_i(x) L_j(t), \tag{1.8}$$

where N_x and N_t are the number of grid points in the x and t directions respectively and $L_i(x)$ is the Lagrange interpolation polynomial given by equation (1.5). Equations (1.8) interpolates $u(x,t)$ at selected points in both the x and t directions defined by

$$\{x_i\} = \left\{\cos\left(\frac{\pi i}{N_x}\right)\right\}_{i=0}^{N_x}, \quad \{t_j\} = \left\{\cos\left(\frac{\pi j}{N_t}\right)\right\}_{j=0}^{N_t}. \tag{1.9}$$

These methods are termed bivariate spectral methods. The values of the time derivatives at the grid points (x_i,t_j), are computed as (for $j = 0,1,2,\ldots,N_t$)

$$\left.\frac{\partial u}{\partial t}\right|_{x=x_i,t=t_j} = \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u(x_p,t_k)L_p(x_i)\frac{dL_k(t_j)}{dt} \tag{1.10}$$

$$= \sum_{k=0}^{N_t} u(x_i,t_k)d_{jk} \tag{1.11}$$

$$= \sum_{k=0}^{N_t} d_{jk}u(x_i,t_k) \tag{1.12}$$

where $d_{jk} = \frac{dL_k(t_j)}{dt}$ is the standard first derivative Chebyshev differentiation matrix of size $(N_t+1)\times(N_t+1)$ as defined by Trefethen [41]. The values of the space derivatives at the grid points (x_i,t_j) $(i = 0,1,2,\cdots,N_x)$ are computed as

$$\left.\frac{\partial u}{\partial x}\right|_{x=x_i,t=t_j} = \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u(x_p,t_k)\frac{dL_p(x_i)}{dx}L_k(t_j) \tag{1.13}$$

$$= \sum_{p=0}^{N_x} u(x_p,t_j)D_{ip} \tag{1.14}$$

$$= \sum_{p=0}^{N_x} D_{ip}u(x_p,t_j), \tag{1.15}$$

where $D_{ip} = \frac{dL_p(x_i)}{dx}$, is the standard first derivative Chebyshev differentiation matrix of size $(N_x+1)\times(N_x+1)$. Similarly, for an nth order space derivative, we have

$$\left.\frac{\partial^n u}{\partial x^n}\right|_{x=x_i,t=t_j} = \sum_{p=0}^{N_x} D_{ip}^n u(x_p,t_j) = \mathbf{D}^n\mathbf{U}_j, \quad i = 0,1,2,\ldots,N_x, \tag{1.16}$$

where the vector $\mathbf{U}_j$ is defined as

$$\mathbf{U}_j = [u_j(x_0), u_j(x_1), \ldots, u_j(x_{N_x})]^T. \tag{1.17}$$

and the superscript T denotes matrix transpose. In this thesis, we develop seven new methods; all of which can be used to solve nonlinear parabolic partial differential equations and systems of equations. The first new method we develop is termed the bivariate spectral quasilinearization method (BSQLM), which is suitable, among other applications, for solving equations arising from fluid mechanics. The method uses Chebyshev spectral collocation, bivariate Lagrange interpolation polynomials together with quasilinearization techniques. The nonlinear parabolic equations are first linearized using the quasilinearization method. The Chebyshev spectral collocation method with Lagrange interpolation polynomials are applied independently in space and time domains. The BSQLM algorithm is presented for a general setting, where it can be used to solve any pth order nonlinear evolution equation. The next method introduced in this thesis is the bivariate spectral quasilinearization method (BSQLM) which is used to solve coupled system of nonlinear equations. Linearization is done for all the variable per equation using the quasilinearization technique

The third method, the bivariate spectral relaxation method (BSRM) has been developed specifically for nonlinear systems of partial differential equations. In this thesis, we seek to improve the performance of the spectral relaxation method (SRM). We will build on the SRM by applying the spectral method to discretize the derivatives in both space and time variables. The proposed approach combines the relaxation scheme of the SRM which uses the Gauss-Seidel approach, bivariate Lagrange interpolation as well as the Chebyshev spectral collocation method.

Another technique, the bivariate spectral local linearization method (BSLLM) is introduced in this thesis. This method is developed for solving a system of n coupled non-similar boundary layer partial differential equations. Linearization is done for one variable per equation using the quasilinearization technique. The system of equations are then solved iteratively using a Jacobi-like relaxation approach.

The fifth technique, the Legendre-Gauss-Lobbatto bivariate pseudospectral quasilinearisation method (LGL-BSQLM) has been developed for nonlinear evolution equations. This method is an improvement of the first method introduced in this thesis, the bivariate spectral quasilinearization

method. The main difference between the two methods is that the LGL-BSQLM uses Legendre-Gauss-Lobatto grid points while the BSQLM uses Chebyshev-Gauss-Lobatto grid points. These techniques perform well in small domains.

If the domains are large, the accuracy of the methods deteriorates. To further improve the accuracy of the methods, the domains are decomposed into smaller sub-intervals. The approach is briefly described in Section 1.2.4.

1.3.2 Multi-domain Bivariate Spectral Methods

Increasing the length of the variables domain decreases the accuracy of the bivariate spectral methods for solving nonlinear partial differential equations. To achieve accurate results, the number of grid points should be increased. Increasing the number of grid points increases computational time and requires more computational resources. The idea is to break down the domain into small sub-domains and solve the differential equations in each sub-domain with a very small interval.

In this thesis, we therefore introduce a domain decomposition method that uses multiple domains, spectral collocation, bivariate Lagrange interpolation polynomials based on Legendre-Gauss-Lobatto grid points together with quasilinearization. Linearity of the evolution partial differential equations is achieved by the quasilinearization method. A pseudospectral collocation method is applied independently both in space and time. The second independent variable's domain is divided into smaller non-overlapping sub-intervals, on which the pseudospectral collocation method is used to solve the partial differential equations. A continuity condition is used to advance the solution across the sub-intervals. This approach is termed the multi-domain Legendre-Gauss-Lobatto based bivariate Lagrange spectral quasilinearization method (MD-LGL-BSQLM). It is applied to evolution nonlinear partial differential equations. The method is further developed to solve systems of nonlinear partial differential equations arising in fluid dynamics. The method for solving systems of nonlinear partial differential equations is termed the multi-domain bivariate spectral quasilinearization method (MD-BSQLM). The BSLLM is also extended to use the multi-domain approach and the method is termed multi-domain bivariate spectral local linearization method (MD-BSLLM).

1.4 Aims and Objectives

Existing numerical methods for nonlinear parabolic partial differential equations and systems of nonlinear partial differential equations converge very slow, use many grid points, and hence use more computational resources to handle differential equations over large domains.

In this thesis, we introduce new pseudospectral methods for solving nonlinear evolution differential equations and systems of nonlinear partial differential equations. Our methods use spectral collocation independently both in space and time. The algorithms are presented in a general form suitable for solving any nonlinear evolution partial differential equations and systems of nonlinear partial differential equations. A number of different algorithms are presented. Among the methods used to determine and validate the accuracy of the numerical schemes, we have compared numerical results with exact solutions, where such exist, or we have used residual error analysis. For comparison, condition numbers of the variable matrices are used.

The main aim of this thesis is to develop new numerical methods for nonlinear parabolic partial differential equations and systems of nonlinear partial differential equations. These numerical methods should converge very fast, use few grid points, use minimal computational resources and handle differential equations over large time domains.

1.5 Thesis outline

This thesis is organized as follows:

- In Chapter 2, the bivariate spectral quasilinearisation method (BSQLM) for nonlinear evolution equations is introduced. The method is presented in a general form for pth order nonlinear evolution equations. The applicability, accuracy and reliability of the BSQLM is confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation. The results of the BSQLM are compared against exact solutions.

- In Chapter 3, the bivariate spectral quasilinearization method (BSQLM) is applied to a system of partial differential equations. The objective is to solve a system of non-similar boundary layer equations that model magnetohydrodynamics forced convection flow

adjacent to a non-isothermal wedge. This problem had been previously solved numerically by Yih using the Keller-Box method [42]. We show that the BSQLM is more accurate than some traditional numerical methods, computationally fast and robust.

- In Chapter 4, the bivariate spectral relaxation method (BSRM) for systems of nonlinear partial differential equations is presented. The spectral method is applied in both space and time, where it is shown to offer spectral accuracy in both variables. This accuracy is achieved with very few grid points compared to the number needed for finite differences; this improves the efficiency of the method. The BSRM is tested on a system of four partial differential equations modeling an unsteady three dimensional magnetohydrodynamic flow and mass transfer in a porous media. Results are compared with previously published results, the spectral relaxation method, the spectral quasilinearization method and the Keller-box method.

- In Chapter 5, the general performance of three bivariate pseudospectral methods is analyzed. The main aim is to compare the general performance of three bivariate pseudospectral methods and present the generalized BSQLM and introduce a generalized bivariate spectral local linearisation method (BSLLM) for a system of n nonlinear system of partial differential equations. In comparing the accuracy and general performance of the three pseudospectral methods, graphs and tables are presented. The results show that the BSLLM is computationally faster than the BSQLM and BSRM.

- In Chapter 6, the a new method termed the Legendre-Gauss-Lobbatto bivariate spectral quasilinearisation method (LGL-BSQLM) for nonlinear evolution partial differential equations is introduced. This method is a modification of the BSQLM method introduced in second chapter of this thesis. The method is presented in a general form for pth order nonlinear evolution equations. It's applicability, accuracy and reliability of the LGL-BSQLM is confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation.

- In Chapter 7, the multi-domain bivariate spectral quasilinearization method (MD-LGL-BSQLM) for nonlinear evolution equations is introduced. The method is presented in a

general form for pth order nonlinear evolution equations over a large time interval. The applicability, accuracy and reliability of the proposed BSQLM is confirmed by solving the modified KdV-Burgers equation, highly nonlinear modified KdV equation, Fisher equation, Burgers-Fisher equation, Burgers-Huxley equation and Fitzhugh-Nagumo equation.

- In Chapter 8, domain decomposition numerical methods for finding solutions of systems of nonlinear coupled non-similar boundary layer partial differential equations over a large time interval are presented. The pseudospectral methods termed the multi-domain bivariate spectral quasilinearization method (MD-BSQLM), and multi-domain bivariate spectral local linearization method (MD-BSLLM), are developed for solving systems of n coupled non-similar boundary layer partial differential equations. The domain is divided into smaller non-overlapping sub-intervals on which the Chebyshev spectral collocation method is used to solve the equations. A continuity condition is used to advance the solution across the sub-intervals. The algorithms presented in this chapter are simple and yet they yield accurate results using few discretization points. The accuracy of the method is validated against the series solution for limiting cases.

- In chapter 9, we conclude the thesis with a summary and suggestions for future work. The references follow thereafter.

Chapter 2

A bivariate Chebyshev spectral collocation quasilinearization method for nonlinear evolution parabolic equations

In this chapter, we present a numerical method for solving higher order nonlinear evolution partial differential equations (NPDEs). The method combines quasilinearization, the Chebyshev spectral collocation method and bivariate Lagrange interpolation. In this chapter, we use the method to solve several nonlinear evolution equations; namely the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation. The results are compared with known exact analytical solutions from literature. Convergence of the method is verified through convergence graphs are and error graphs used to compare the results from this study and the known results from literature. This chapter reflects work we have already published as indicated by publication number 1 (see page *vii*).

2.1 Introduction

Nonlinearity exists everywhere, and in general, nature is nonlinear. Nonlinear evolution partial differential equations arise in many fields of science, particularly in physics, engineering, chemistry, finance and biological systems. They are widely used to describe complex phenomena in various fields of sciences, such as wave propagation phenomena, fluid mechanics, plasma

physics, quantum mechanics, non-linear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, financial industry and numerous areas of mathematical modeling. The development of both numerical and analytical methods for solving complicated, highly nonlinear evolution partial differential equations continues to be an area of interest to scientists whose research aim is to enrich deep understanding of such alluring nonlinear problems.

Innumerable number of methods for obtaining analytical and approximate solutions to nonlinear evolution equations have been proposed. Some of the analytical methods that have been used to solve evolution nonlinear partial differential equations include Adomian's decomposition method [43–45], homotopy analysis method [46–49], tanh-function method [50–52], Haar wavelet method [53–55] and Exp-function method [56–58]. Several numerical methods have been used to solve nonlinear evolution partial differential equations. These include the explicit-implicit method [59], Chebyshev finite difference methods [60], finite difference methods [61], finite element methods [62] and pseudospectral methods [63, 64].

Some drawbacks of approximate analytic methods include slow convergence, particularly for large time ($t > 1$). They may also be cumbersome to use as some involve manual integration of approximate series solutions and hence it is difficult to find closed solutions sometimes. On the other hand, some numerical methods may not work in some cases, for example when the required solution has to be found near a singularity. Certain numerical methods, for example finite differences require many grid points to achieve good accuracy and hence require a lot of computer memory and computational time. Conventional first-order finite difference methods may result in monotonic and stable solutions, but they are strongly dissipative causing the solution of the strongly convective partial differential equations to become smeared out and often grossly inaccurate. On the other hand, higher order difference methods are less dissipative but are prone to numerical instabilities.

Spectral methods have been used successfully in many different fields in the sciences and engineering because of their ability to give accurate solutions of differential equations. Khater [37] applied the Chebyshev spectral collocation method to solve Burgers type of equations in space and finite differences to approximate the time derivative. The Chebyshev spectral collocation method has been used together with the fourth-order Runge-Kutta method to solve the nonlinear PDEs in this study. The Chebyshev spectral collocation is first applied to the

NPDE and this yields a system of ordinary differential equations, which are solved using the fourth-order Runge-Kutta method. Olmos [65], Javidi [38, 39], Dehghan [40], Driscoll [66], solved the Fisher, Burgers-Fisher, Burgers-Huxley, and Fitzhugh-Nagumo equations respectively using a combination of the Chebyshev spectral collocation method and fourth-order Runge-Kutta method. Darvishi [67, 68] solved the KdV and the Burgers-Huxley equations using a combination of the Chebyshev spectral collocation method and Darvishi's preconditioning. Jacobs [69] and Tohidi [70] used spectral collocation directly for solving linear partial differential equations. Accuracy will be compromised if they implement their approach in solving nonlinear partial differential equations since they use Kronecker multiplication.

Chebyshev spectral methods are defined everywhere in the computational domain. Therefore, it is easy to get an accurate value of the function under consideration at any point of the domain, beside the collocation points. This property is often exploited, in particular to get a significant graphic representation of the solution, making apparent the possible oscillations due to a wrong approximation of the derivative. Spectral collocation methods are easy to implement and are adaptable to various problems, including variable coefficient and nonlinear differential equations. The error associated with the Chebyshev approximation is $\mathscr{O}(1/N^r)$ where N refers to the truncation and r is connected to the number of continuous derivatives of the function. The interest in using Chebyshev spectral methods in solving nonlinear PDEs stems from the fact that these methods require less grid points to achieve accurate results. They are computational efficient compared to traditional methods like finite difference and finite element methods. Chebyshev spectral collocation method have been used in conjunction with additional methods which may have their own drawbacks. Here we provide an alternative method that is not dependent on another method to approximate the solution.

The main objective of this chapter is to introduce a new method that uses Chebyshev spectral collocation, bivariate Lagrange interpolation polynomials together with quasilinearisation techniques. The nonlinear evolution equations are first linearized using the quasilinearisation method. The Chebyshev spectral collocation method with Lagrange interpolation polynomials are applied independently in space and time variables of the linearized evolution partial differential equation. This new method is termed bivariate interpolated spectral quasilinearisation method (BSQLM). We present the BSQLM algorithm in a general setting, where it can be used to solve any rth order nonlinear evolution equations. The applicability, accuracy and reliability

of the proposed BSQLM is confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation. The results of the BSQLM are compared against known exact solutions that have been reported in the scientific literature. It is observed that the method achieves high accuracy with relatively fewer spatial grid points. It also converges fast to the exact solution and approximates the solution of the problem in a computationally efficient manner with simulations completed in fractions of a second in all cases. Tables are generated to show the order of accuracy of the method and time taken to compute the solutions. It is observed that as the number of grid points is increased, the error decreases. Error graphs and graphs showing the excellent agreement of the exact and analytical solutions for all the nonlinear evolution equations are also presented.

This chapter is organized as follows. In Section 2, we introduce the BSQLM algorithm for a general non-linear evolution PDE. In Section 3, we describe the application of the BSQLM to selected test problems. The numerical simulations and results are presented in Section 4. Finally, we conclude in Section 5.

2.2 Bivariate Spectral Quasilinearization Method (BSQLM)

In this section we introduce the Bivariate Spectral Quasilinearization Method (BSQLM) for finding solutions to non-linear evolution PDEs. Without loss of generality, we consider non-linear PDEs of the form,

$$\frac{\partial u}{\partial \tau} = H\left(u, \frac{\partial u}{\partial \eta}, \frac{\partial^2 u}{\partial \eta^2}, \cdots, \frac{\partial^n u}{\partial \eta^n}\right), \text{with the physical region } \tau \in [0,T],\ \eta \in [a,b] \qquad (2.1)$$

where n is the order of differentiation, $u(\eta,\tau)$ is the required solution and H is a non-linear operator which contains all the spatial derivatives of u. The given physical region, $\tau \in [0,T]$ is converted to the region $t \in [-1,1]$ using the linear transformation $\tau = T(t+1)/2$ and $\eta \in [a,b]$ is converted to the region $x \in [-1,1]$ using the linear transformation

$$\eta = \frac{1}{2}(b-a)x + \frac{1}{2}(b+a).$$

Equation (2.1) can be expressed as

$$\frac{\partial u}{\partial t} = H\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \cdots, \frac{\partial^n u}{\partial x^n}\right), \quad t \in [-1,1], \quad x \in [-1,1] \tag{2.2}$$

The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form (1.8), which interpolates $u(x,t)$ at selected points in both the x and t directions defined by equation (1.9). The choice of the Chebyshev-Gauss-Lobatto grid points (1.9), ensures that there is a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. Before expressing equation (2.2) in linear form, it is convenient to split H into its linear and nonlinear components and rewrite the governing equation in the form,

$$F[u, u', \cdots, u^{(n)}] + G[u, u', \cdots, u^{(n)}] - \dot{u} = 0, \tag{2.3}$$

where the dot and primes denote the time and space derivatives, respectively, F is a linear operator and G is a non-linear operator. Assuming that the difference $u_{r+1} - u_r$ and all it's space derivative is small, we first approximate the non-linear operator G using the linear terms of the Taylor series and hence

$$G[u, u', \cdots, u^{(n)}] \approx G[u_r, u'_r, \cdots, u_r^{(n)}] + \sum_{k=0}^{n} \frac{\partial G}{\partial u^{(k)}} \left(u_{r+1}^{(k)} - u_r^{(k)}\right) \tag{2.4}$$

where r and $r+1$ denote previous and current iterations respectively. We remark that this quasi-linearization method (QLM) approach is a generalisation of the Newton-Raphson method and was first proposed by Bellman and Kalaba [71] for solving nonlinear boundary value problems. Equation (2.4) can be expressed as

$$G[u, u', \cdots, u^{(n)}] \approx G[u_r, u'_r, \cdots, u_r^{(n)}] + \sum_{k=0}^{n} \phi_{k,r}[u_r, u'_r, \cdots, u_r^{(n)}] u_{r+1}^{(k)} - \sum_{k=0}^{n} \phi_{k,r}[u_r, u'_r, \cdots, u_r^{(n)}] u_r^{(k)} \tag{2.5}$$

where

$$\phi_{k,r}[u_r, u'_r, \cdots, u_r^{(n)}] = \frac{\partial G}{\partial u^{(k)}}[u_r, u'_r, \cdots, u_r^{(n)}]. \tag{2.6}$$

Substituting equation (2.5) into equation (2.3), we get

$$F[u_{r+1}, u'_{r+1}, \cdots, u^{(n)}_{r+1}] + \sum_{k=0}^{n} \phi_{k,r} u^{(k)}_{r+1} - \dot{u}_{r+1} = R_r[u_r, u'_r, \cdots, u^{(n)}_r] \tag{2.7}$$

where

$$R_r[u_r, u'_r, \cdots, u^{(n)}_r] = \sum_{k=0}^{n} \phi_{k,r} u^{(k)}_r - G[u_r, u'_r, \cdots, u^{(n)}_r].$$

A crucial step in the implementation of the solution procedure is the evaluation of the time derivative at the grid points t_j ($j = 0, 1, \ldots, N_t$) and the spatial derivatives at the grid points x_i ($i = 0, 1, \ldots, N_x$). The values of the time derivatives at the Chebyshev-Gauss-Lobatto points (x_i, t_j), are computed using equation (1.12) for $j = 0, 1, 2, \ldots, N_t$. The values of the space derivatives at the Chebyshev-Gauss-Lobatto points (x_i, t_j) for $i = 0, 1, 2, \cdots, N_x$ are computed using equation (1.16). Substituting (1.16) into (2.7) we get

$$F[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \cdots, \mathbf{U}^{(n)}_{r+1,j}] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t} d_{jk} \mathbf{U}_{r+1,k} = R_r[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \cdots, \mathbf{U}^{(n)}_{r,j}] \tag{2.8}$$

for $j = 0, 1, 2, \ldots, N_t$, where

$$\mathbf{U}^{(n)}_{r+1,j} = \mathbf{D}^n \mathbf{U}_{r+1,j}, \quad \mathbf{\Phi}_{k,r} = \begin{bmatrix} \phi_{k,r}(x_0, t_j) & & & \\ & \phi_{k,r}(x_1, t_j) & & \\ & & \ddots & \\ & & & \phi_{k,r}(x_{N_x}, t_j) \end{bmatrix} \tag{2.9}$$

The initial condition for equation (2.2) corresponds to $\tau_{N_t} = -1$ and hence we express equation (2.8) as

$$F[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \cdots, \mathbf{U}^{(n)}_{r+1,j}] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t - 1} d_{jk} \mathbf{U}_{r+1,k} = \mathbf{R}_j \tag{2.10}$$

where

$$\mathbf{R}_j = R_r[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \cdots, \mathbf{U}^{(n)}_{r,j}] + d_{jN_t} \mathbf{U}_{N_t}, \quad j = 0, 1, 2, \ldots, N_t - 1.$$

Equation (2.10) can be expressed as the following $N_t(N_x+1)\times N_t(N_x+1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \tag{2.11}$$

where

$$A_{i,i} = F[\mathbf{I}, \mathbf{D}, \cdots, \mathbf{D}^{(n)}] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r}\mathbf{D}^{(k)} - d_{i,i}\mathbf{I} \tag{2.12}$$

$$A_{i,j} = -d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \tag{2.13}$$

and $\mathbf{I}$ is the identity matrix of size $(N_x+1)\times(N_x+1)$. Solving equation (2.11) gives an approximate value of $u(x,t)$.

2.3 Numerical experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (2.2) with exact solutions. In order to determine the level of accuracy of the BSQLM approximate solution, at a particular time level, in comparison with the exact solution we report maximum error which is defined by

$$E_N = \max_r \left\{ |u(x_r,t) - \tilde{u}(x_r,t)| \, , : 0 \leq r \leq N \right\}, \tag{2.14}$$

where $\tilde{u}(x_r,t)$ is the approximate solution and is the $u(x_r,t)$ exact solution at the time level t.

Example 1: We consider the generalized Burgers-Fisher equation [72]

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta}), \tag{2.15}$$

with initial condition

$$u(x,0) = \left\{ \frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}x\right) \right\}^{\frac{1}{\delta}}, \tag{2.16}$$

and exact solution

$$u(x,t) = \left\{ \frac{1}{2} + \frac{1}{2} \tanh\left(\frac{-\alpha\delta}{2(\delta+1)} \left[x - \left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha} \right) t \right] \right) \right\}^{\frac{1}{\delta}}, \tag{2.17}$$

where α, β and δ are parameters. For illustration purposes, these parameters are chosen to be $\alpha = \beta = \delta = 1$ in this paper. The linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' + u\,, \quad G(u) = -uu' - u^2. \tag{2.18}$$

We first linearize the nonlinear operator G. We approximate G using the equation

$$G \approx G[u_r, u'_r, u''_r] + \sum_{k=0}^{2} \phi_{k,r} u_{r+1}^{(k)} - \sum_{k=0}^{2} \phi_{k,r} u_r^{(k)}. \tag{2.19}$$

The coefficients are given by

$$\phi_{0,r} = \frac{\partial G}{\partial u}[u_r, u'_r, u''_r] = -(u'_r + 2u_r), \tag{2.20}$$

$$\phi_{1,r} = \frac{\partial G}{\partial u'}[u_r, u'_r, u''_r] = -u_r, \tag{2.21}$$

$$\phi_{2,r} = \frac{\partial G}{\partial u''}[u_r, u'_r, u''_r] = 0, \tag{2.22}$$

and

$$R_r = \sum_{k=0}^{2} \phi_{k,r} u_r^{(k)} - G[u_r, u'_r, u''_r] = -u_r^2 - u_r u'_r. \tag{2.23}$$

Therefore, the linearized equation can be expressed as

$$u''_{r+1} + \phi_{1,r} u'_{r+1} + \phi_{0,r} u_{r+1} + u_{r+1} - \dot{u} = R_r. \tag{2.24}$$

Applying the spectral method both in x and t, and initial condition, we get

$$\mathbf{D}^2 \mathbf{U}_{r+1,i} + \mathbf{\Phi}_{1,r} \mathbf{D} \mathbf{U}_{r+1,i} + \mathbf{\Phi}_{0,r} \mathbf{U}_{r+1,i} + \mathbf{U}_{r+1,i} - 2 \sum_{j=0}^{N_t - 1} d_{ij} \mathbf{U}_{r+1,j} = \mathbf{R}_i. \tag{2.25}$$

Equation (2.25) can be expressed as

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \tag{2.26}$$

where

$$A_{i,i} = \mathbf{D}^2 + \mathbf{\Phi}_{1,r}^{(i)}\mathbf{D} + \mathbf{\Phi}_{0,r}^{(i)} + (1-2d_{i,i})\mathbf{I}, \tag{2.27}$$

$$A_{i,j} = -2d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \tag{2.28}$$

$$\mathbf{R}_i = R_r + 2d_{iN_t}\mathbf{U}_{r,N_t}. \tag{2.29}$$

The boundary conditions are implemented in the first and last row of the matrices A_{ij} and the column vectors $\mathbf{R}_i$ for $i = 0, 1, \cdots, N_t - 1$ and $j = 0, 1, \cdots, N_t - 1$. The procedure for finding the variable coefficients ϕ_i and matrices for the remaining examples is similar.

Example 2: We consider the Fisher equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u(1-u), \tag{2.30}$$

subject to the initial condition

$$u(x,0) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x}\right)^2}, \tag{2.31}$$

and exact solution [73]

$$u(x,t) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}x - 5\alpha t/6}\right)^2}, \tag{2.32}$$

where α is a constant. The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. For this example, the appropriate linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' + \alpha u\,, \quad G(u) = -\alpha u^2. \tag{2.33}$$

Example 3: Consider the Fitzhugh-Nagumo equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(u-\alpha)(1-u) \tag{2.34}$$

with initial condition

$$u(x,0) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}}\right)\right]. \tag{2.35}$$

This equation has the exact solution [74]

$$u(x,t) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}} + \frac{2\alpha-1}{4}t\right)\right], \tag{2.36}$$

where α is a parameter. In this example the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \alpha u\,, \quad G(u) = (1+\alpha)u^2 - u^3. \tag{2.37}$$

Example 4: Consider the Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} u_x = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta})(u^{\delta} - \gamma), \tag{2.38}$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study) and $\gamma \in (0,1)$. The exact solution subject to the initial condition

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r-\alpha}x\right], \tag{2.39}$$

is reported in [75, 76] as

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r-\alpha}(x-ct)\right], \tag{2.40}$$

where

$$r = \sqrt{\alpha^2 + 8\beta} \quad \text{and} \quad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4} \tag{2.41}$$

The general solution (2.40) was reported in [77, 78]. In this example the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \beta\gamma u\,, \quad G(u) = -\alpha u u' + \beta(1+\gamma)u^2 - \beta u^3. \tag{2.42}$$

Example 5: We consider the modified KdV-Burgers equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 u}{\partial x^2} - 6u^2\frac{\partial u}{\partial x} \tag{2.43}$$

subject to the initial condition

$$u(x,0) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6}\right) \tag{2.44}$$

and exact solution [79]

$$u(x,t) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6} - \frac{t}{27}\right) \tag{2.45}$$

The modified KdV-Burgers equation describes various kinds of phenomena such as a mathematical model of turbulence and the approximate theory of flow through a shock wave traveling in viscous fluid [80]. For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''' - u''\,, \quad G(u) = -6u'u^2. \tag{2.46}$$

Example 6: We consider the high nonlinear modified KdV equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} + \left(\frac{\partial u}{\partial x}\right)^2 - u^2 \tag{2.47}$$

subject to the initial condition

$$u(x,0) = \frac{1}{2} + \frac{e^{-x}}{4} \tag{2.48}$$

and exact solution

$$u(x,t) = \frac{1}{t+2} + \frac{e^{-(x+t)}}{(t+2)^2} \tag{2.49}$$

For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''' \,, \quad G(u) = (u')^2 - u^2. \tag{2.50}$$

2.4 Results and Discussion

In this section we present the numerical solutions obtained using the BSQLM algorithm. The number of collocation points in the space x variable used to generate the results is $N_x = 10$ in all cases. Similarly, the number of collocation points in the time t variable used is $N_t = 10$ in all cases. It was found that sufficient accuracy was achieved using these values in all numerical simulations. The results in Tables 2.1 - 2.12 and Figs. 2.1 - 2.12 were obtained using MATLAB 2013b. Figs. 2.13 - 2.18 were obtained using Mathematica 9 to clearly demonstrate that the method has converged.

In Tables 2.1 - 2.6 we give the maximum errors between the exact and BSQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation respectively, at $t \in [0.1, 1]$. The results were computed in the space domain $x \in [0, 1]$. To give a sense of the computational efficiency of the method, the computational time to generate the results is also given. Tables 2.1 - 2.6 clearly show the accuracy of the method. The accuracy is seen to improve with an increase in the number of collocation points N_x. It is remarkable to note that accurate results with errors of order up to 10^{-14} are obtained using very few collocation points in both the x and t variables $N_t \leq 10, N_x \leq 10$. This is a clear indication that the BSQLM is powerful method that is appropriate in solving nonlinear evolution PDEs. We remark, also, that the BSQLM is computationally fast as accurate results are generated in a fraction of a second in all the examples considered in this work.

Table 2.1 Maximum errors E_N for the Fisher equation when α = 1 using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	1.986e-008	1.119e-011	7.398e-013	7.171e-013
0.2	3.934e-008	3.121e-011	1.552e-012	1.561e-012
0.3	5.577e-008	4.864e-011	1.004e-012	1.005e-012
0.4	6.997e-008	6.802e-011	7.895e-013	8.124e-013
0.5	8.107e-008	7.971e-011	1.088e-012	1.027e-012
0.6	8.891e-008	8.560e-011	8.805e-013	7.847e-013
0.7	9.344e-008	8.953e-011	6.418e-013	6.463e-013
0.8	9.431e-008	8.759e-011	6.199e-013	6.164e-013
0.9	9.178e-008	8.325e-011	3.978e-013	3.695e-013
1.0	8.787e-008	7.421e-011	7.988e-014	5.596e-014
CPU Time (sec)	0.019942	0.025988	0.027756	0.029436

Table 2.2 Maximum errors E_N for the Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	1.142e-007	1.369e-010	5.891e-012	6.143e-012
0.2	1.178e-007	1.373e-010	9.570e-012	1.013e-011
0.3	1.186e-007	1.479e-010	1.489e-011	1.512e-011
0.4	1.069e-007	9.450e-011	1.703e-011	1.702e-011
0.5	9.030e-008	7.944e-011	5.283e-012	5.736e-012
0.6	6.963e-008	6.618e-011	1.639e-011	1.626e-011
0.7	4.638e-008	1.579e-011	1.362e-011	1.364e-011
0.8	2.457e-008	4.030e-011	3.934e-012	3.852e-012
0.9	2.028e-008	6.006e-011	4.466e-012	4.727e-012
1.0	3.147e-008	7.708e-011	7.757e-013	7.261e-013
CPU Time (sec)	0.010152	0.015387	0.018163	0.019564

Table 2.3 Maximum errors E_N for the Fitzhugh-Nagumo equation when α = 1 using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	5.719e-007	1.196e-009	2.367e-012	9.881e-014
0.2	6.193e-007	1.299e-009	2.761e-012	3.952e-014
0.3	6.662e-007	1.463e-009	3.259e-012	8.216e-014
0.4	6.779e-007	1.448e-009	3.341e-012	8.094e-014
0.5	6.920e-007	1.526e-009	3.587e-012	5.063e-014
0.6	7.019e-007	1.573e-009	3.729e-012	3.775e-014
0.7	6.933e-007	1.516e-009	3.660e-012	8.915e-014
0.8	6.828e-007	81.535e-009	3.635e-012	7.594e-014
0.9	6.765e-007	1.528e-009	3.519e-012	3.242e-013
1.0	6.687e-007	1.490e-009	3.405e-012	1.688e-013
CPU Time (sec)	0.024281	0.024901	0.026810	0.032389

Table 2.4 Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.75$, $\beta = 1$, $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	2.217e-006	8.482e-009	2.166e-011	7.822e-014
0.2	2.596e-006	9.369e-009	2.536e-011	1.184e-013
0.3	2.859e-006	1.073e-008	3.201e-011	1.049e-013
0.4	3.001e-006	1.112e-008	3.652e-011	9.426e-014
0.5	3.137e-006	1.213e-008	4.262e-011	1.510e-013
0.6	3.270e-006	1.311e-008	4.842e-011	2.127e-013
0.7	3.367e-006	1.359e-008	5.289e-011	1.230e-013
0.8	3.467e-006	1.438e-008	5.803e-011	1.549e-013
0.9	3.562e-006	1.504e-008	6.260e-011	3.063e-013
1.0	3.640e-006	1.559e-008	6.674e-011	2.951e-013
CPU Time (sec)	0.023822	0.024901	0.02685	0.032806

Table 2.5 Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	1.803e-007	3.419e-010	4.449e-013	1.572e-013
0.2	2.614e-007	4.347e-010	5.049e-013	5.992e-014
0.3	2.717e-007	4.677e-010	5.532e-013	8.128e-013
0.4	2.009e-007	3.663e-010	4.771e-013	6.158e-013
0.5	2.580e-007	4.410e-010	7.518e-013	2.555e-013
0.6	2.653e-007	4.606e-010	8.738e-013	5.756e-013
0.7	2.248e-007	4.039e-010	6.210e-013	2.393e-013
0.8	2.572e-007	4.476e-010	5.432e-013	6.812e-013
0.9	2.436e-007	4.351e-010	6.111e-013	6.287e-013
1.0	8.275e-008	3.721e-010	7.569e-013	1.087e-013
CPU Time (sec)	0.015646	0.021226	0.030159	0.035675

Table 2.6 Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.1	7.788e-005	3.553e-007	7.601e-010	2.080e-010
0.2	1.153e-004	4.000e-007	5.684e-010	1.189e-010
0.3	1.011e-004	3.739e-007	4.471e-010	4.503e-010
0.4	3.926e-005	1.785e-007	6.544e-010	4.987e-010
0.5	6.727e-005	2.342e-007	2.638e-010	1.528e-010
0.6	6.065e-005	2.207e-007	4.565e-010	4.568e-010
0.7	2.511e-005	1.105e-007	4.749e-010	3.748e-010
0.8	4.074e-005	1.427e-007	1.062e-010	1.604e-010
0.9	2.386e-005	1.018e-007	2.343e-010	8.114e-011
1.0	1.440e-006	7.256e-008	1.436e-009	1.513e-011
CPU Time (sec)	0.020609	0.021241	0.030617	0.032816

In Tables 2.7 - 2.12 we give the maximum errors of the BSQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation respectively, at selected values of $t = 2$ for different collocation points, N_t, in the t-variable. The results in Tables 2.7 - 2.12 were computed on the space domain $x \in [0,1]$. We note that the accuracy does not detoriate when $t > 1$ for this method as is often the case with numerical schemes such as finite differences.

Table 2.7 Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	1.119e-011	7.398e-013	8.266e-013	3.808e-014
0.4	3.121e-011	1.552e-012	7.378e-013	3.780e-014
0.6	4.864e-011	1.004e-012	3.402e-012	7.283e-014
0.8	6.802e-011	7.895e-013	1.118e-012	3.714e-014
1.0	7.971e-011	1.088e-012	1.473e-012	1.691e-013
1.2	8.560e-011	8.805e-013	2.611e-012	3.119e-013
1.4	8.953e-011	6.418e-013	6.671e-012	1.796e-013
1.6	8.759e-011	6.199e-013	1.118e-011	1.097e-013
1.8	8.325e-011	3.978e-013	7.515e-013	6.273e-014
2.0	7.421e-011	7.988e-014	3.682e-012	2.311e-013
CPU Time (sec)	0.013542	0.022967	0.023792	0.024758

Table 2.8 Maximum errors E_N for the Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	1.223e-007	1.400e-008	1.402e-008	1.094e-012
0.4	1.145e-007	1.919e-008	1.918e-008	3.919e-012
0.6	9.192e-008	2.082e-008	2.085e-008	1.953e-012
0.8	2.293e-008	1.793e-008	1.793e-008	6.340e-013
1.0	2.395e-008	1.337e-008	1.339e-008	2.381e-012
1.2	5.778e-008	1.954e-008	1.930e-008	1.005e-011
1.4	6.045e-008	1.620e-008	1.620e-008	3.535e-012
1.6	5.244e-008	7.218e-009	7.345e-009	5.765e-012
1.8	4.395e-008	6.828e-009	6.784e-009	3.983e-012
2.0	2.944e-008	9.406e-010	8.820e-010	3.812e-012
CPU Time (sec)	0.019942	0.025988	0.027756	0.029436

Table 2.9 Maximum errors E_N for the Fitzhugh-Nagumo equation when α = 1 using $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	6.326e-007	1.311e-009	2.886e-012	1.131e-012
0.4	6.721e-007	1.467e-009	3.310e-012	1.564e-012
0.6	7.140e-007	1.602e-009	3.617e-012	1.936e-012
0.8	6.730e-007	1.496e-009	4.707e-012	1.196e-012
1.0	6.660e-007	1.487e-009	3.675e-012	1.264e-012
1.2	6.449e-007	1.366e-009	1.897e-012	1.727e-012
1.4	5.690e-007	1.083e-009	2.972e-012	1.200e-012
1.6	4.931e-007	8.010e-010	1.519e-012	8.590e-013
1.8	3.986e-007	4.658e-010	1.068e-012	6.790e-013
2.0	2.904e-007	2.968e-010	1.592e-012	1.770e-013
CPU Time (sec)	0.041048	0.049629	0.055008	0.053863

Table 2.10 Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.5$, $\beta = 1$, $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	2.866e-006	1.119e-008	3.670e-011	1.150e-012
0.4	3.401e-006	1.420e-008	5.744e-011	1.638e-012
0.6	3.814e-006	1.687e-008	7.426e-011	1.958e-012
0.8	3.915e-006	1.729e-008	8.171e-011	7.002e-013
1.0	3.938e-006	1.738e-008	8.157e-011	1.267e-012
1.2	3.808e-006	1.624e-008	7.687e-011	1.710e-012
1.4	3.456e-006	1.527e-008	6.965e-011	5.109e-013
1.6	3.230e-006	1.349e-008	5.535e-011	8.203e-013
1.8	2.925e-006	1.078e-008	3.598e-011	8.294e-013
2.0	2.497e-006	7.505e-009	2.265e-011	9.726e-014
CPU Time (sec)	0.023822	0.024901	0.02685	0.032806

Table 2.11 Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	2.137e-007	3.820e-010	4.846e-013	9.998e-013
0.4	2.480e-007	4.267e-010	5.596e-013	8.775e-013
0.6	2.691e-007	4.676e-010	6.565e-013	2.054e-012
0.8	2.214e-007	3.979e-010	8.776e-013	1.168e-012
1.0	2.538e-007	4.463e-010	9.650e-013	8.410e-013
1.2	2.650e-007	4.680e-010	7.450e-013	5.113e-013
1.4	2.383e-007	4.296e-010	7.500e-013	1.110e-012
1.6	2.568e-007	4.572e-010	9.704e-013	2.837e-013
1.8	2.520e-007	4.529e-010	7.443e-013	5.353e-013
2.0	2.370e-007	4.438e-010	2.719e-013	8.849e-013
CPU Time (sec)	0.062066	0.081646	0.080718	0.10775

Table 2.12 Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$

$t \setminus N_x$	4	6	8	10
0.2	1.986e-008	1.119e-011	7.398e-013	7.171e-013
0.4	8.010e-005	3.577e-007	3.902e-008	1.979e-010
0.6	7.235e-005	2.549e-007	2.016e-008	4.899e-010
0.8	6.284e-005	1.663e-007	1.155e-007	2.679e-010
1.0	1.642e-005	1.620e-007	1.243e-007	2.474e-010
1.2	2.753e-005	1.073e-007	1.073e-007	1.679e-010
1.4	3.738e-006	8.971e-008	8.598e-008	4.788e-011
1.6	1.223e-005	2.153e-008	2.503e-008	2.941e-011
1.8	5.836e-006	2.986e-008	9.127e-009	5.177e-011
2.0	9.310e-006	6.548e-008	7.277e-008	1.453e-009
CPU Time (sec)	0.020609	0.021241	0.030617	0.032816

Figures 2.1 - 2.6 show a comparison of the analytical and approximate solutions of the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation. The approximate solutions are in excellent agreement with the analytical solutions, and this demonstrates the accuracy of the algorithm presented in this study.

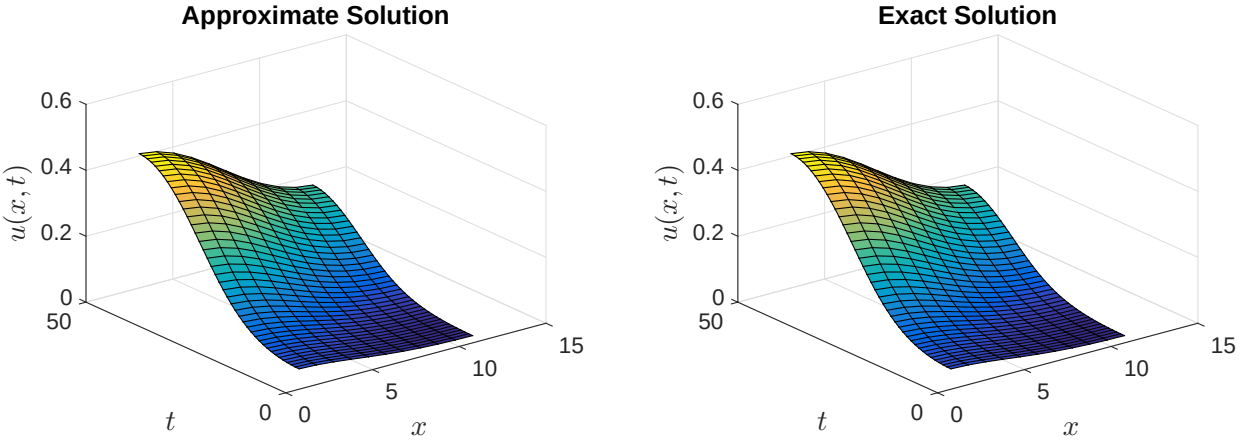

Fig. 2.1 Approximate and Exact solutions of the Fisher equation.

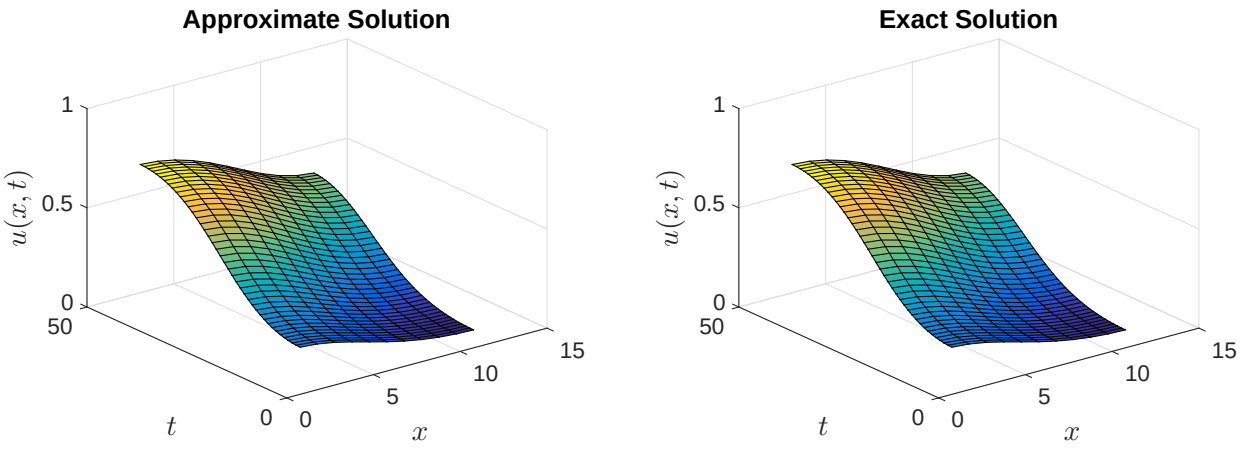

Fig. 2.2 Approximate and Exact solutions of the Burgers-Fisher equation.

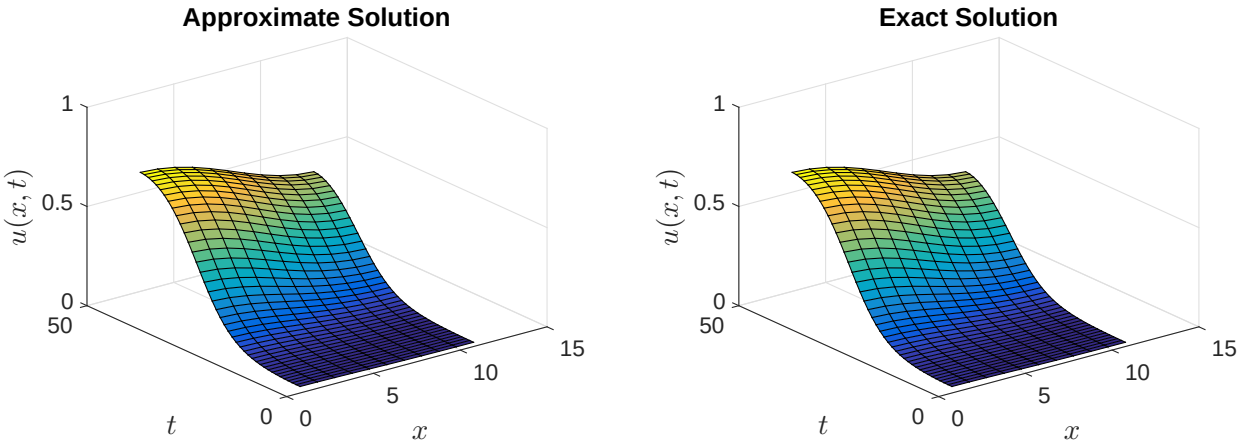

Fig. 2.3 Approximate and Exact solutions of the Burgers-Huxley equation.

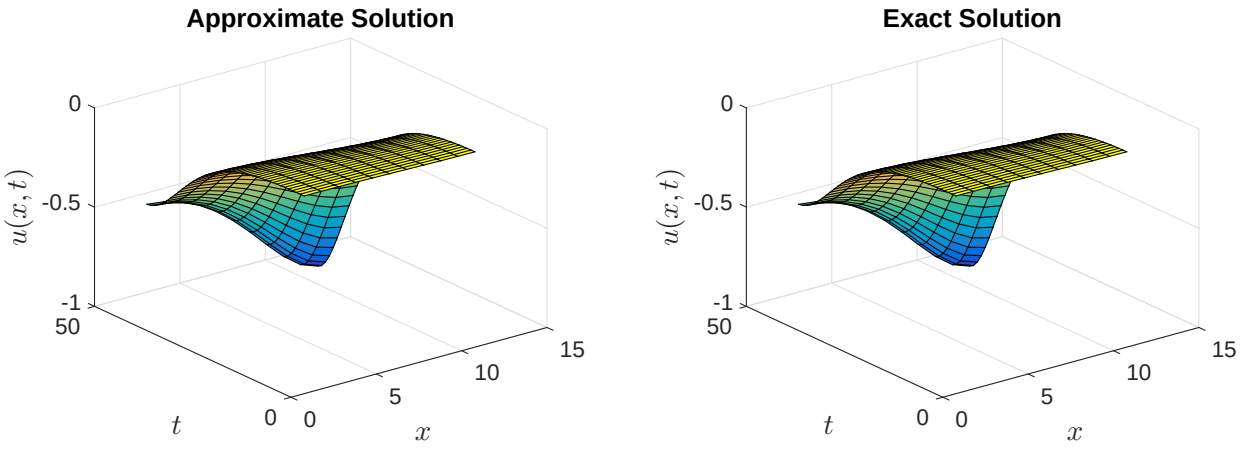

Fig. 2.4 Approximate and Exact solutions of the Fitzhugh-Nagumo equation.

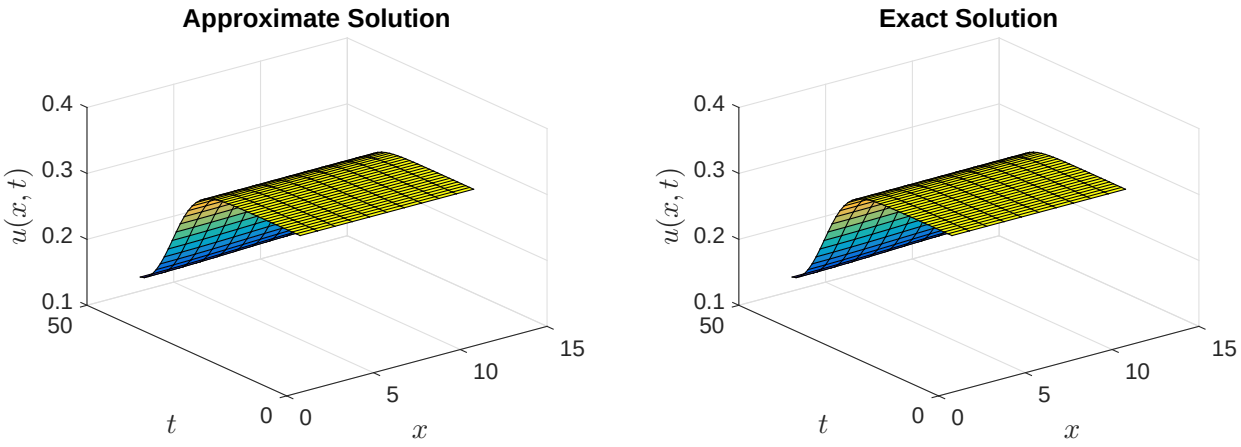

Fig. 2.5 Approximate and Exact solutions of the modified KdV-Burgers equation.

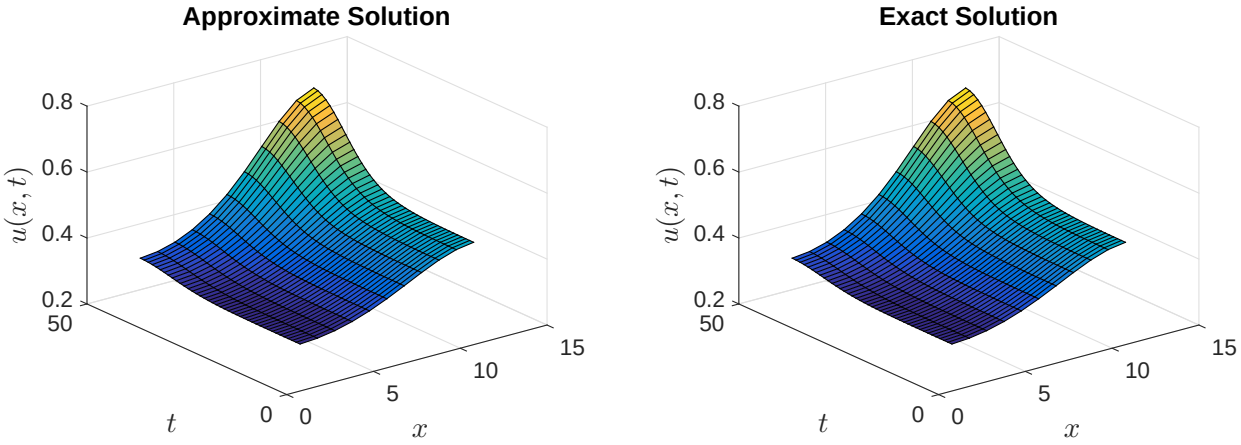

Fig. 2.6 Approximate and Exact solutions of the modified KdV equation.

In Figs. 2.7 - 2.12 we present error analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation. The errors presented here are very small and thus confirming the accuracy of the BSQLM method over the entire domains of the nonlinear evolution equations.

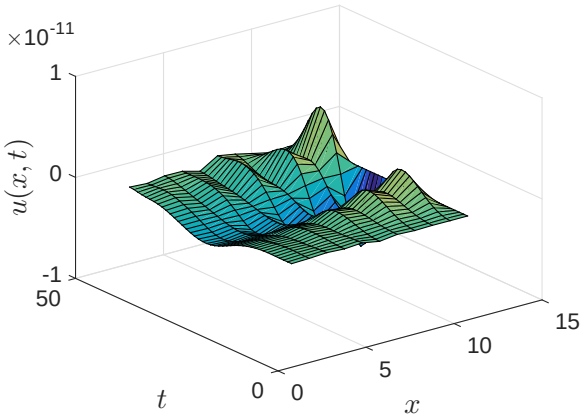

Fig. 2.7 Fishers equation error graph

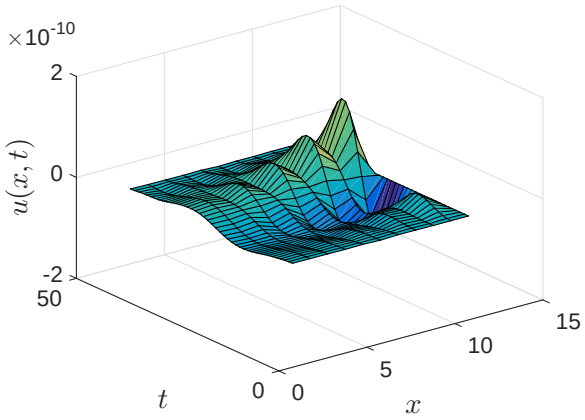

Fig. 2.8 Burger-Fishers equation error graph

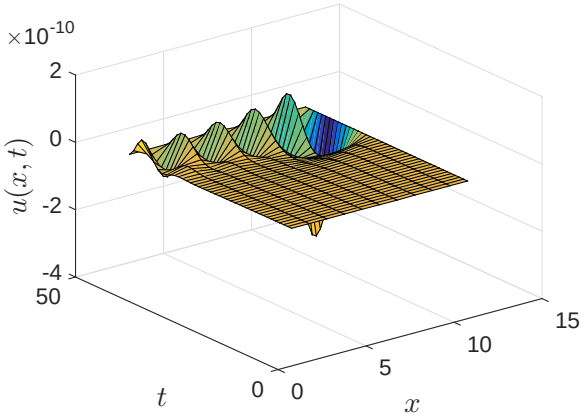

Fig. 2.9 Fitzhugh-Nagumo equation error graph

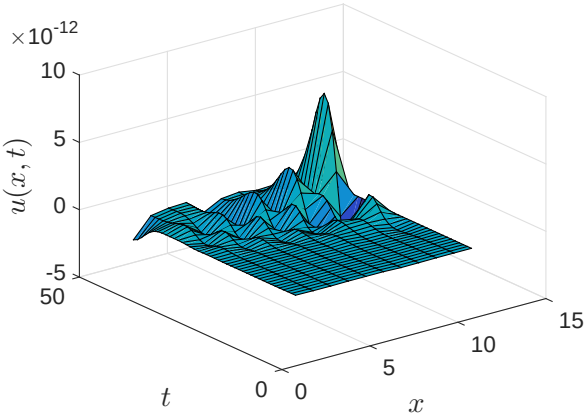

Fig. 2.10 Burgers-Huxley equation error graph

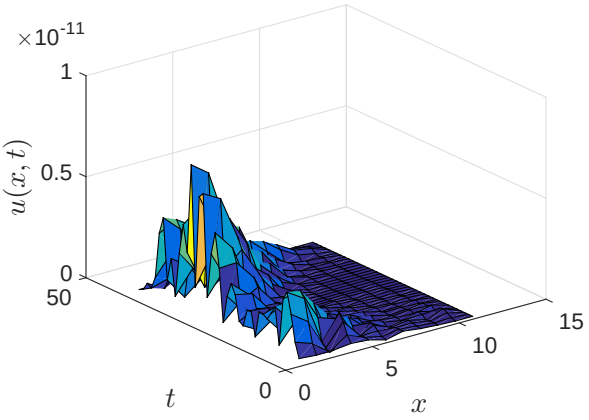

Fig. 2.11 Modified KdV-Burgers equation error graph

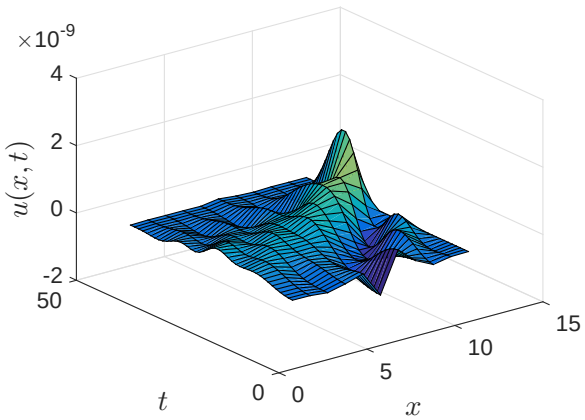

Fig. 2.12 Modified KdV equation error graph

In Figs. 2.13 - 2.18 convergence analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively. T The figures present a variation of the error norm at a fixed values of time ($t = 1$) with iterations of the BSQLM scheme. It can be seen that, in almost all the examples considered, the iteration scheme takes about 3 or 4 iterations to converge fully. Beyond the point where full convergence is reached, error norm levels off and does not improve with an increase in the number of iterations. This plateau level gives an estimate of the maximum error that can be achieved when using the proposed method with a certain number of collocation points. It is worth remarking that the accuracy of the method depends on the number of collocation points in both the x and t directions. The results from Figs. 2.13 - 2.18 clearly demonstrate that the BSQLM is second order accurate.

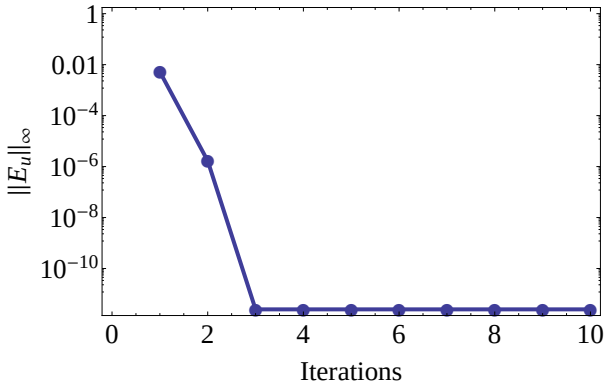

Fig. 2.13 Fisher equation convergence graph

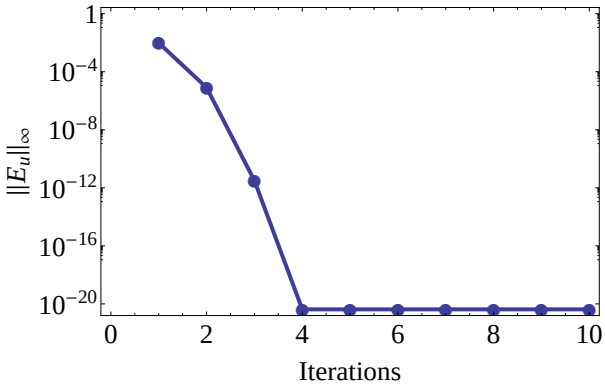

Fig. 2.14 Burger-Fisher convergence graph

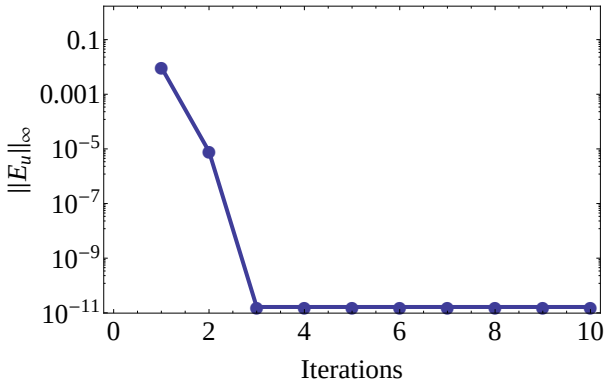

Fig. 2.15 Fitzhugh-Nagumo equation convergence graph

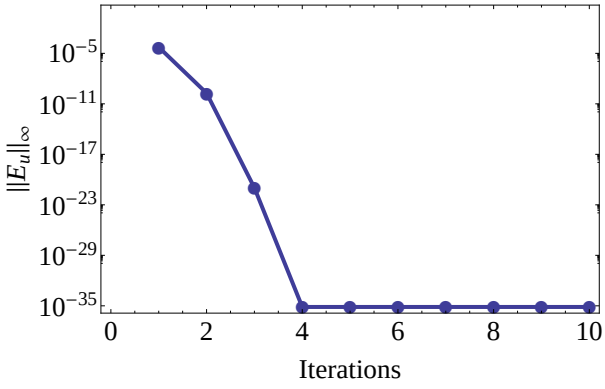

Fig. 2.16 Burgers-Huxley equation convergence graph

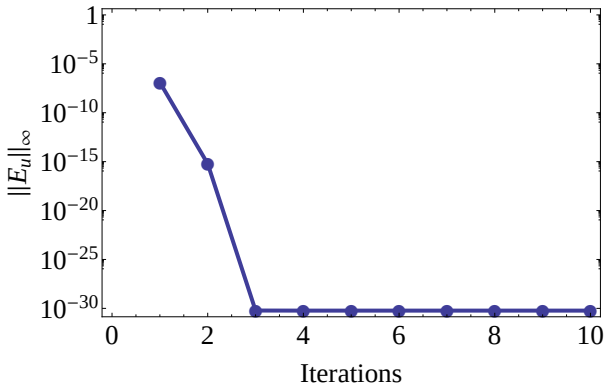

Fig. 2.17 Modified KdV-Burger equation convergence graph

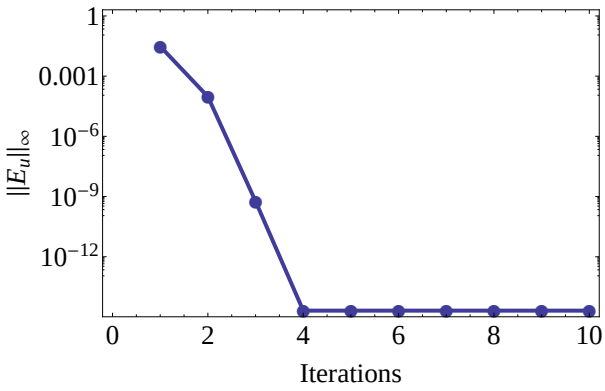

Fig. 2.18 Modified KdV equation convergence graph

2.5 Conclusion

In this chapter, we presented a new Chebyshev collocation spectral method for solving general nonlinear evolution partial differential equations. The bivariate spectral quasilinearization method (BSQLM) was developed by combining elements of the quasilinearization method and Chebyshev spectral collocation with bivariate Lagrange interpolation. The main goal of this analysis was to assess the accuracy, robustness and effectiveness of the method in solving nonlinear partial differential equations. Numerical simulations were conducted on the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation. It is evident from the study that the BSQLM gives accurate results in a computationally efficient manner. Furthermore evidence shows that the BSQLM gives solutions that are uniformly accurate and valid over large intervals in the space and time domains. The apparent success of the method can be attributed to the use of the Chebyshev spectral collocation method with bivariate Lagrange interpolation in space and time for differentiating. This modification is superior to previously used methods in that it converges faster, uses few grid points to achieve accurate results and uses few computational resources compared to traditional methods like finite differences. This work thus contributes to the existing body of literature on quasilinearization tools for solving complex nonlinear partial differential equations. Having shown here that the BSQLM is suitable for solving single equations, in the next chapter, we develop it for solving system of nonlinear differential equations.

Chapter 3

Bivariate spectral collocation based quasilinearization method for solving non-similar boundary layer equations

In this chapter, we present an innovative technique for solving nonlinear systems of partial differential equations, which has quadratic convergence rate. We demonstrate use of the method by solving a system of nonlinear partial differential equations that belong to a class of non-similar boundary layer equations. The equations used here model the problem of magnetohydrodynamic forced convection flow adjacent to a non-isothermal wedge. The results from the method are validated against published results. Convergence analysis and grid independence tests are also conducted to establish the accuracy, convergence rate and validity of the proposed method. Computational order of convergence is proved numerically by tabulated results.

3.1 Introduction

Many physical processes may be described by systems of complex nonlinear differential equations. Very few of these equations have closed form solutions, so finding their solutions poses significant challenges. Nevertheless, there exist a wide range of both numerical and analytical techniques to find approximate solutions for such equations. However, these methods are not universally applicable; in some cases they may suffer from instabilities or may not be sufficiently accurate or computationally efficient. Devising new and more efficient algorithms is always a

challenge and an interesting research objective. The problem considered in this chapter relates to the flow of a viscous incompressible magnetohydrodyanamic fluid, which finds applications in many engineering processes.

Various numerical techniques have been developed to solve a class of non-similar boundary layer equations. The nonlinear equations that describe the fluid flow model in this chapter have been solved previously by Yih [42] using the Keller-box method. The Keller-box method, or its variants, have been widely used to solve other non-similar boundary layer equations; for example, Cebeci and Bradshaw [81], Aydin and Kaya [82], Prasad et al. [83] and Afify and Bazid [84]. The Keller-box method is an implicit finite difference scheme that is second order accurate, both in space and time. It permits the step sizes in time and space variables to be nonuniform. This makes it efficient and appropriate for the solution of non-similar boundary layer equations. The main disadvantage of the method is the expensive computational effort per time step. In this regard, higher order derivatives must be replaced at the start by first order derivatives so that an nth order non-similar boundary layer equation is then expressed as a system of n first-order equations [85]. The size of the coefficient matrix from discretization increases due to the introduction of the derivatives as unknowns. This in turn implies that many grid points are needed to achieve accurate results. All of these steps increase the computational cost and time.

Other finite difference schemes have been used to solve non-similar boundary layer equations. Watanabe [86] and Watanabe and Pop [87] used the backward finite differences while Watanabe and Pop [88] used a Gregory-Newton finite difference method. Ariel [89] used a combination of finite differences with quasilinearization. Chamkha et al. [90] and Chamkha and Rashad [91] used an implicit iterative tridiagonal finite difference method.

While explicit methods break down if the time step is too large, as was described in Section 1.2.1, implicit finite difference methods are unconditionally stable. However, the extra accuracy and efficiency of an implicit method may be counterbalanced by the extensive computing time required to solve the linear system using an iterative method. Hence the extra accuracy can be seen as a drawback for the method. The Runge-Kutta-Fehlberg scheme with a shooting technique has been used to find numerical approximations of non-similar boundary layer equations [92–95]. However, for stiff problems, explicit Runge-Kutta methods are very inefficient.

Spectral methods give accurate solutions of differential equations and have been used successfully in many scientific fields. As was described previously in Section 1.2.2, Chebyshev spectral methods are defined everywhere in the computational domain. Thus, it is easy to get an accurate value of the function at any point of the domain. Spectral collocation methods are simple to implement and easily adapted to different problems, including variable coefficient and nonlinear differential equations. The interest in using Chebyshev spectral methods in solving nonlinear differential equation stems from these methods requiring few grid points to achieve accurate results and are computationally efficient when solving problems with smooth solutions. Spectral methods coupled with finite differences have been used to solve nonlinear partial differential equations [96] in space and finite differences in time. However, applying finite differences in time compromises the accuracy and computationally efficiency of spectral methods. Nevertheless, spectral methods have been applied successfully to solve nonlinear evolution equations in both space and time, as was described in Chapter 2, and published by Motsa, Magagula and Sibanda [97]. We called the method the bivariate spectral quasilinearization method (BSQLM).

The objective of this chapter, is to now extend the use of the bivariate spectral quasilinearization method (BSQLM) to solve a system of non-similar boundary layer equations that model magnetohydrodynamics forced convection flow adjacent to a non-isothermal wedge. As already noted, Yih [42] had previously solved this problem numerically using the Keller-box method. In our method we will first linearize a system of coupled nonlinear equations, collocate and solve the resulting matrix iteratively. We will compare results from the method with those from Yih [42] and evaluate its accuracy, computational efficiency and robustness.

3.2 Bivariate Spectral Quasilinearization Method

In this section we consider non-similar boundary layer equations that model the problem of magnetohydrodynamics forced convection flow adjacent to a non-isothermal wedge. The

equations are given in dimensionless form as [42]

$$\frac{\partial^3 f}{\partial \eta^3}+\beta f\frac{\partial^2 f}{\partial \eta^2}+m\left[1-\left(\frac{\partial f}{\partial \eta}\right)^2\right]+\xi\left(1-\frac{\partial f}{\partial \eta}\right)=(1-m)\xi\left(\frac{\partial f}{\partial \eta}\frac{\partial^2 f}{\partial \eta\partial \xi}-\frac{\partial^2 f}{\partial \eta^2}\frac{\partial f}{\partial \xi}\right),$$
$$\frac{1}{Pr}\frac{\partial^2 \theta}{\partial \eta^2}+\beta f\frac{\partial \theta}{\partial \eta}-2m\theta\frac{\partial f}{\partial \eta}+Ec\left[\left(\frac{\partial^2 f}{\partial \eta^2}\right)^2-(m+\xi)\frac{\partial f}{\partial \eta}+\xi\left(\frac{\partial f}{\partial \eta}\right)^2\right] \quad (3.1)$$
$$=(1-m)\xi\left(\frac{\partial f}{\partial \eta}\frac{\partial \theta}{\partial \xi}-\frac{\partial \theta}{\partial \eta}\frac{\partial f}{\partial \xi}\right),$$

subject to the boundary conditions

$$f(0,\xi)=0,\quad \frac{\partial f}{\partial \eta}(0,\xi)=0,\quad \frac{\partial f}{\partial \eta}(\infty,\xi)=1,\quad \theta(0,\xi)=1,\quad \theta(\infty,\xi)=0, \quad (3.2)$$

where m, Pr, Ec, ξ, η, θ and f are the dimensionless pressure gradient parameter, Prandtl number, Eckert number, magnetic parameter, pseudo-similarity variable, dimensionless temperature function, dimensionless stream function respectively and $\beta=(1+m)/2$. In the analysis of boundary layer flow problems, quantities of physical interest are the skin friction, and the local Nusselt number given in dimensionless form (see Yih [42]) as

$$C_f Re_x^{1/2}=2\frac{\partial^2 f}{\partial \eta^2}(\xi,0), \quad (3.3)$$

$$Nu_x=-Re_x^{1/2}\frac{\partial \theta}{\partial \eta}(\xi,0), \quad (3.4)$$

respectively, where Re_x is the local Reynolds number, C_f is the local friction coefficient and Nu_x is the local Nusselt number. We now find the approximate solutions of equations (3.1 - 3.2). We apply the quasilinearization method by assuming that the difference $f_{r+1}-f_r$, and all its derivatives, are small. The quasilinearization method is the Taylor series expansion about a point of the first two linear terms. It was introduced by Bellman and Kalaba [71]. Applying the quasilinearization method to equation (3.1) gives

$$f'''_{r+1}+a_{1,r}(\eta,\xi)f''_{r+1}+a_{2,r}(\eta,\xi)f'_{r+1}+a_{3,r}(\eta,\xi)f_{r+1}+a_{4,r}(\eta,\xi)\frac{\partial f'_{r+1}}{\partial \xi}$$
$$+a_{5,r}(\eta,\xi)\frac{\partial f_{r+1}}{\partial \xi}=a_{6,r}(\eta,\xi), \quad (3.5)$$
$$\theta''_{r+1}+\alpha_{1,r}(\eta,\xi)\theta'_{r+1}+\alpha_{2,r}(\eta,\xi)\theta_{r+1}+\alpha_{3,r}(\eta,\xi)\frac{\partial \theta_{r+1}}{\partial \xi}=\alpha_{4,r}(\eta,\xi),$$

where the primes denote differentiation with respect to η and

$$a_{1,r}(\eta,\xi) = \beta f_r + (1-m)\xi\frac{\partial f_r}{\partial \xi}, \quad a_{2,r}(\eta,\xi) = -2mf_r' - \xi - (1-m)\xi\frac{\partial f_r'}{\partial \xi},$$
$$a_{3,r}(\eta,\xi) = \beta f_r'', \quad a_{4,r}(\eta,\xi) = -(1-m)\xi f_r', \quad a_{5,r}(\eta,\xi) = (1-m)\xi f_r'',$$
$$a_{6,r}(\eta,\xi) = \beta f_r f_r'' - m(f_r')^2 - m - \xi - (1-m)\xi\left(f'\frac{\partial f_r'}{\partial \xi} - f''\frac{\partial f_r}{\partial \xi}\right)$$
$$\alpha_{1,r}(\eta,\xi) = \beta Pr f_r + Pr(1-m)\xi\frac{\partial f_r}{\partial \xi}, \quad \alpha_{2,r}(\eta,\xi) = -2mPrf_r',$$
$$\alpha_{3,r}(\eta,\xi) = -Pr(1-m)\xi f_r', \quad \alpha_{4,r}(\eta,\xi) = -EcPr\left[(f_r'')^2 - (m+\xi)f_r' + \xi(f_r')^2\right]$$

The solution of the now linear differential equation (3.5) is obtained by approximating the solution $f(\eta,\xi)$ using a Lagrange polynomial $F(\eta,\xi)$ which interpolates $f(\eta,\xi)$ at selected collocation points,

$$0 = \xi_0 < \xi_1 < \xi_2 < \cdots < \xi_{N_\xi} = L_\xi.$$

Thus, the approximation for $f(\eta,\xi)$ has the form

$$f(\eta,\xi) \approx \sum_{j=0}^{N_\xi} F(\eta,\xi_j)L_j(\xi) = \sum_{j=0}^{N_\xi} F_j(\eta)L_j(\xi), \tag{3.6}$$

where $F_j(\eta) \equiv F(\eta,\xi_j)$ and $L_j(\xi)$ is the characteristic Lagrange cardinal polynomial defined by equation (1.5) satisfying the properties in equation (1.6). The solutions $F_j(\eta)$ are obtained by substituting (3.6) in (3.5) and requiring the equation to be satisfied exactly at the points ξ_i, $i = 0,1,2,\ldots,N_\xi$. For the derivatives of the Lagrange polynomial with respect to ξ to be computable analytically, it is convenient to transform the interval $\xi \in [0,L_\xi]$ to $\zeta \in [-1,1]$ then choose Chebyshev-Gauss-Lobatto points (1.9), as collocation points. After using a linear transformation to transform ξ to the new variable ζ, the derivative of f' with respect to ξ at the collocation points ζ_j is computed as

$$\left.\frac{\partial f'}{\partial \xi}\right|_{\xi=\xi_i} = 2\sum_{j=0}^{N_\xi} F_j'(\eta)\frac{dL_j}{d\zeta}(\zeta_i) = \sum_{j=0}^{N_\xi} \mathbf{d}_{i,j}F_j'(\eta), \quad i = 0,1,2,\ldots,N_\xi, \tag{3.7}$$

where

$$d_{i,j} = \frac{dL_j}{d\zeta}(\zeta_i), \quad i = 0,1,\ldots,N_\xi \tag{3.8}$$

are entries of the standard Chebyshev differentiation matrix (see, for example [41, 98]), and $\mathbf{d} = (2/L_\xi)d$. Applying the collocation at ξ_i in equations (3.5) gives

$$\begin{aligned} F'''_{r+1,i}(\eta) + a^{(i)}_{1,r}F''_{r+1,i}(\eta) + a^{(i)}_{2,r}F'_{r+1,i}(\eta) + a^{(i)}_{3,r}F_{r+1,i}(\eta) + a^{(i)}_{4,r}\sum_{j=0}^{N_\xi}\mathbf{d}_{i,j}F'_{r+1,j}(\eta) \\ + a^{(i)}_{5,r}\sum_{j=0}^{N_\xi}\mathbf{d}_{i,j}F_{r+1,j}(\eta) = a^{(i)}_{6,r}, \\ \Theta''_{r+1,i}(\eta) + \alpha^{(i)}_{1,r}\Theta'_{r+1,i}(\eta) + \alpha^{(i)}_{2,r}\Theta_{r+1,i}(\eta) + \alpha^{(i)}_{3,r}\sum_{j=0}^{N_\xi}\mathbf{d}_{i,j}\Theta_{r+1,j} = \alpha^{(i)}_{4,r}, \end{aligned} \tag{3.9}$$

where $a^{(i)}_{k,r} \equiv a_{k,r}(\eta,\xi_i)$ $(k = 1,2,3,4,5,6)$ and $\alpha^{(i)}_{s,r} \equiv \alpha_{s,r}(\eta,\xi_i)$ $(s = 1,2,3,4)$. Consequently, for each ξ_i, equations (3.9) form a linear ordinary equation with variable coefficients. To solve equations (3.9), we apply the Chebyshev spectral collocation independently in the η direction by choosing $N_\eta + 1$ Chebyshev-Gauss-Lobatto points $0 = \eta_0 < \eta_1 < \cdots < \eta_{N_\eta} = \eta_e$, where η_e is a finite value that is chosen to be sufficiently large to approximate the conditions at infinity. Again, before implementing the collocation, the interval $\eta \in [0,\eta_e]$ is transformed into $\tau \in [-1,1]$ using a linear transformation. Thus, the collocation points are chosen as $\tau_j = \cos\left(\dfrac{j\pi}{N_\eta}\right)$. The derivatives with respect to η are defined is terms of the Chebyshev differentiation matrix as

$$\left.\frac{d^pF_{r+1,i}}{d\eta^p}\right|_{\eta=\eta_j} = \left(\frac{2}{\eta_e}\right)^p\sum_{k=0}^{N_\eta}D^p_{j,k}F_{r+1,i}(\tau_k) = \mathbf{D}^p\mathbf{F}_{r+1,i}, \tag{3.10}$$

where p is the order of the derivative, $\mathbf{D} = (2/\eta_e)[D_{j,k}]$ $(j,k = 0,1,2,\ldots,N_\eta)$ with $[D_{j,k}]$ being an $(N_\eta+1)\times(N_\eta+1)$ Chebyshev derivative matrix, and the vector $\mathbf{F}_{r+1,i}$ is defined as

$$\mathbf{F}_{r+1,i} = [F_{r+1,i}(\tau_0), F_{r+1,i}(\tau_1), \ldots, F_{r+1,i}(\tau_{N_\eta})]^T.$$

Thus applying equation (3.10) in equations (3.9) gives

$$\begin{aligned} \mathbf{A}^{(i)}\mathbf{F}_{r+1,i} - 2\xi_i(1-\xi_i)\sum_{j=0}^{N_\xi}d_{i,j}\mathbf{D}\mathbf{F}_{r+1,j} = \mathbf{R}^{(i)}_1, \\ \mathbf{B}^{(i)}\boldsymbol{\Theta}_{r+1,i} + \boldsymbol{\alpha}^{(i)}_{3,r}\sum_{j=0}^{N_\xi}d_{i,j}\boldsymbol{\Theta}_{r+1,j} = \boldsymbol{\Gamma}^{(i)}_1 \end{aligned} \tag{3.11}$$

where

$$\begin{aligned}
\mathbf{A}^{(i)} &= \mathbf{a}_{0,r}^{(i)}\mathbf{D}^3 + \mathbf{a}_{1,r}^{(i)}\mathbf{D}^2 + \mathbf{a}_{2,r}^{(i)}\mathbf{D} + \mathbf{a}_{3,r}^{(i)}, \\
\mathbf{B}^{(i)} &= \boldsymbol{\alpha}_{0,r}^{(i)}\mathbf{D}^2 + \boldsymbol{\alpha}_{1,r}^{(i)}\mathbf{D} + \boldsymbol{\alpha}_{2,r}^{(i)}, \\
\mathbf{R}_1^{(i)} &= \mathbf{a}_{6,r}^{(i)}, \\
\mathbf{\Gamma}_1^{(i)} &= \boldsymbol{\alpha}_{4,r}^{(i)}.
\end{aligned} \tag{3.12}$$

$\mathbf{a}_{k,r}^{(i)}$ $(k = 0,1,2,3,4,5)$, $\boldsymbol{\alpha}_{s,r}^{(i)}$ $(s = 0,1,2,3)$ are diagonal matrices with vectors $[a_{k,r}^{(i)}(\tau_0),\ a_{k,r}^{(i)}(\tau_1),\cdots,$ $a_{k,r}^{(i)}(\tau_{N_x})]^T$, $[\alpha_{s,r}^{(i)}(\tau_0), \alpha_{s,r}^{(i)}(\tau_1),\cdots,\alpha_{s,r}^{(i)}(\tau_{N_x})]^T$ placed on the main diagonal, $\mathbf{a}_{6,r}^{(i)} = [a_{6,r}^{(i)}(\tau_0), a_{6,r}^{(i)}(\tau_1),\cdots,a_{6,r}^{(i)}(\tau_{N_x})]^T$, and $\boldsymbol{\alpha}_{4,r}^{(i)} = [\alpha_{4,r}^{(i)}(\tau_0), \alpha_{4,r}^{(i)}(\tau_1),\cdots,\alpha_{4,r}^{(i)}(\tau_{N_x})]^T$. After imposing the boundary conditions, for each $i = 0,1,\ldots,N_\xi$ equations (3.11) can be written in matrix form as

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_\xi} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_\xi} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_\xi,0} & A_{N_\xi,1} & \cdots & A_{N_\xi,N_\xi} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{r+1,0} \\ \mathbf{F}_{r+1,1} \\ \vdots \\ \mathbf{F}_{r+1,N_\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_1^{(0)} \\ \mathbf{R}_1^{(1)} \\ \vdots \\ \mathbf{R}_1^{(N_\xi)} \end{bmatrix}, \tag{3.13}$$

$$\begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,N_\xi} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,N_\xi} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\xi,0} & B_{N_\xi,1} & \cdots & B_{N_\xi,N_\xi} \end{bmatrix} \begin{bmatrix} \mathbf{\Theta}_{r+1,0} \\ \mathbf{\Theta}_{r+1,1} \\ \vdots \\ \mathbf{\Theta}_{r+1,N_\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{\Gamma}_1^{(0)} \\ \mathbf{\Gamma}_1^{(1)} \\ \vdots \\ \mathbf{\Gamma}_1^{(N_\xi)} \end{bmatrix}, \tag{3.14}$$

where

$$A_{i,i} = \mathbf{A}^{(i)} + \mathbf{a}_{4,r}^{(i)}\mathbf{d}_{i,i}\mathbf{D} + \mathbf{a}_{5,r}^{(i)}\mathbf{d}_{i,i}, \quad i = 0,1,\ldots,N_\xi, \tag{3.15}$$

$$A_{i,j} = \mathbf{a}_{4,r}^{(i)}\mathbf{d}_{i,j}\mathbf{D} + \mathbf{a}_{5,r}^{(i)}\mathbf{d}_{i,j},, \quad \text{when } i \neq j, \tag{3.16}$$

$$B_{i,i} = \mathbf{B}^{(i)} + \boldsymbol{\alpha}_{3,r}^{(i)}\mathbf{d}_{i,i}, \tag{3.17}$$

$$B_{i,j} = \boldsymbol{\alpha}_{3,r}^{(i)}\mathbf{d}_{i,i}. \tag{3.18}$$

Equations (3.13) and (3.14) can be expressed in the form

$$\mathbf{AF} = \mathbf{R}, \tag{3.19}$$

$$\mathbf{B\Theta} = \mathbf{\Gamma}. \tag{3.20}$$

The solutions are found by multiplying by the inverses of **A** and **B** respectively to get

$$\mathbf{F} = \mathbf{A}^{-1}\mathbf{R}, \tag{3.21}$$

$$\mathbf{\Theta} = \mathbf{B}^{-1}\mathbf{\Gamma}. \tag{3.22}$$

3.3 Results and Discussion

In this section we present the numerical solutions obtained using the BSQLM algorithm. The results presented in this section were generated using MATLAB. The number of collocation points in the space η variable used to generate the results is $N_\eta = 60$ in Table 3.1 through Table 3.4. The number of collocation points in the magnetic parameter ξ variable used is $10 \leq N_\xi \leq 30$ in all cases. It was found that sufficient accuracy was achieved using these values in all numerical simulations. In Tables 3.1 through 3.3, we compare our results with those of Yih [42]. The results are in excellent agreement. The time to compute the approximate solutions by the Bivariate spectral quasilinearization method is displayed to give an indication of the computational speed. The time to compute the approximate solutions using the Bivariate spectral quasilinearization method is less than three seconds for all cases which is remarkable since traditional numerical methods require more computational time. We remark that in Yih's article [42], the stopping criteria and time steps were not provided and hence, we were not able to compare the time to compute the same results using the Keller-Box method.

Table 3.1 Comparison of the numerical values of the skin friction $f''(0,0)$ for various values of m.

$m \setminus N_\xi$	10	15	20	25	Yih[42]
-0.05	0.213484	0.213484	0.213484	0.213484	0.213484
0.0	0.332057	0.332057	0.332057	0.332057	0.332057
1/3	0.757448	0.757448	0.757448	0.757448	0.757448
1.0	1.232588	1.232588	1.232588	1.232588	1.232588
Time (sec)	0.181133	0.469017	0.908688	1.532474	

Table 3.1 shows the numerical values of the skin friction $f''(0,0)$ for various values of m. The numerical values obtained using the BSQLM method agrees exactly with those obtained by Yih[42] using

the finite differences based method, the Keller-Box method. We also note that the BSQLM approximates the skin friction within a fraction of a second.

Table 3.2 Numerical values of the skin friction $f''(0,\xi)$ at different values of ξ when $m = 1$.

$\xi \setminus N_\xi$	10	15	20	25	Yih[42]
0	1.232588	1.232588	1.232588	1.232588	1.232588
1	1.585336	1.585329	1.585331	1.585331	1.585331
4	2.346667	2.346663	2.346663	2.346663	2.346663
9	3.240943	3.240950	3.240950	3.240950	3.240950
25	5.147966	5.147966	5.147966	5.147966	5.147964
CPU Time (sec)	0.448792	0.83394	1.410784	2.250584	

Table 3.2 shows the numerical values of the skin friction $f''(0,\xi)$ at different values of ξ when $m = 1$. The numerical values obtained using the BSQLM method are in agreement with those obtained by Yih [42] using the Keller-Box method. It should be noted that the BSQLM approximates the skin friction within a fraction of a second.

Table 3.3 Numerical values of the Nusselt number $-\theta'(0,\xi)$ at different values of ξ when $m = 0$, $Pr = Ec = 1$.

$\xi \setminus N_\xi$	10	15	20	25	Yih[42]
0.0	0.166029	0.166029	0.166029	0.166029	0.166029
0.5	0.201426	0.201411	0.201411	0.201411	0.201452
1.0	0.216787	0.216786	0.216786	0.216786	0.216814
1.5	0.226296	0.226302	0.226302	0.226302	0.226323
2.0	0.232981	0.232981	0.232981	0.232981	0.232998
CPU Time (sec)	0.15710	0.350761	0.834859	1.496277	

Table 3.3 shows the numerical values of the Nusselt number $-\theta'(0,\xi)$ at different values of ξ when $m = 0$, $Pr = Ec = 1$. The numerical values obtained using the BSQLM method are in agreement with those obtained by Yih[42] using the Keller-Box method. The BSQLM approximates the skin friction within a fraction of a second.

Table 3.4 Numerical values of the Nusselt number $-\theta'(0,\xi)$ at different values of ξ when $m = 1/2$,and $Pr = Ec = 1$.

$\xi \setminus N_\xi$	10	15	20	25	30
0	0.720626	0.720626	0.720626	0.720626	0.720626
0.5	0.759480	0.759480	0.759480	0.759480	0.759480
1	0.786931	0.786931	0.786931	0.786931	0.786931
1.5	0.808036	0.808036	0.808036	0.808036	0.808036
2	0.825096	0.825096	0.825096	0.825096	0.825096
CPU Time(sec)	0.239541	0.468491	0.865754	1.488947	2.17121

Table 3.4 shows the numerical values of the Nusselt number $-\theta'(0,\xi)$ at different values of ξ when $m = 0.5$, $Pr = Ec = 1$. In this case, we use the grid independence test to show that our method can generate approximate values of the Nusselt number with any m in the range of the values of m. We also note that the BSQLM approximates the skin friction within few seconds. We now show the order of accuracy of the method. We start by defining the order of convergence of a method.

Theorem 3.1 (Order of Convergence)**.** *A method is convergent with convergence rate c if*

$$\lim_{k\to\infty} \frac{||e_{k+1}||_\infty}{||e_k||_\infty^c} = V \tag{3.23}$$

where V is a finite positive constant. The convergence rate c can be calculated from the ratio

$$\frac{e_{k+2}}{e_{k+1}} = \left(\frac{e_{k+1}}{e_k}\right)^c \tag{3.24}$$

and hence the convergence rate c is given by

$$c = \frac{\ln(e_{k+2}/e_{k+1})}{\ln(e_{k+1}/e_k)}, \tag{3.25}$$

where e_{k+2}, e_{k+1} and e_k are the norms computed using three consecutive iterations. The error is approximated at a given iteration level by the equation

$$e_k = ||f_{k+1} - f_k||_\infty \tag{3.26}$$

Bellman and Kalaba [71] observed that the quasilinearization method converges quadratically, that is $c = 2$. The results below confirm that the BSQLM converges quadratically. Table 3.5 shows the convergence rate values of $f(\eta,\xi)$ at different values of N_η, when $N_\xi = 8$, $m = 0$, and $Pr = Ec = 1$. Table 3.6 shows the convergence rate values of $\theta(\eta,\xi)$ at different values of N_η, when $N_\xi = 8$, $m = 0$, and $Pr = Ec = 1$. The values in Table 3.5 and Table 3.6 show that the BSQLM converges quadratically.

Table 3.5 Convergence Rate values of $f(\eta,\xi)$ at different values of N_η when $N_\xi = 8$, $m = 0$, and $Pr = Ec = 1$.

	Convergence Rate			
Iterations \ N_η	15	30	45	60
3	2.0100	2.0000	2.0000	2.0000
4	2.0000	2.0000	2.0000	2.0000
5	1.9800	2.0000	2.0000	2.0000
6	2.0000	2.0000	2.0000	2.0000
7	2.0000	1.9900	2.0000	2.0000
8	2.0000	2.0000	1.9800	1.9800

Table 3.6 Convergence Rate values of $\theta(\eta,\xi)$ at different values of N_η when $N_\xi = 8$, $m = 0$, and $Pr = Ec = 1$.

	Convergence Rate			
Iterations \ N_η	15	30	45	60
3	1.9500	1.9600	1.9600	1.9600
4	1.9500	1.9900	1.9900	1.9900
5	2.0500	2.0300	2.0300	2.0300
6	2.0300	2.0100	2.0100	2.0100
7	2.0000	2.0000	2.0000	2.0000
8	1.9800	1.9900	1.9900	1.9900

In order to determine the level of accuracy of the BSQLM approximate solution, at a particular time level,we report maximum error which is defined by

$$E_f = \max_i \left\{ ||\mathbf{F}_{r+1,i} - \mathbf{F}_{r,i}| \, |_\infty, \, : 0 \leq i \leq N_\xi \right\}, \tag{3.27}$$

$$E_\theta = \max_i \left\{ ||\mathbf{\Theta}_{r+1,i} - \mathbf{\Theta}_{r,i}| \, |_\infty, \, : 0 \leq i \leq N_\xi \right\}, \tag{3.28}$$

where $\mathbf{F}_{r+1,i}$ and $\mathbf{F}_{r,i}$ are the approximate solutions obtained by equation (3.21) at the current and previous time steps respectively.

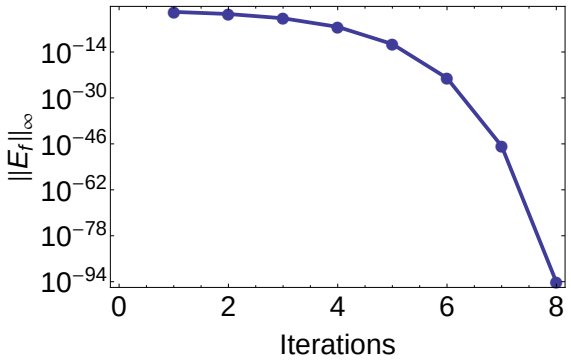

Fig. 3.1 E_f vs iterations for all η and ξ

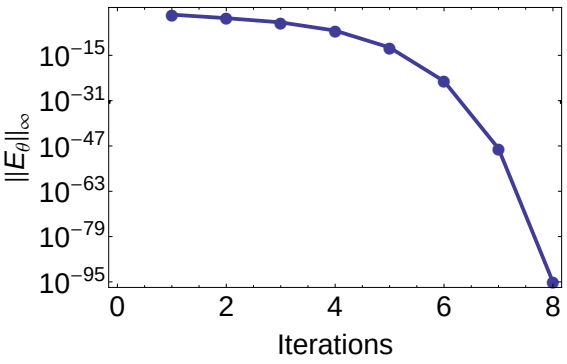

Fig. 3.2 E_θ vs iterations for all η and ξ

Figure 3.1 and Figure 3.2 show the convergence graphs of f and θ respectively. The BSQLM method converges fully after eight iterations. In Figure 3.1, the BSQLM method converges to 10^{-94} after eight iterations. We note that it converges quadratically and hence the parabolic shape of the of the curve. This is in agreement with the the results in Tables 3.5 and 3.6 and the theory that the quasilinearization method converges quadratically as observed by Bellman and Kalaba [71]. In Figure 3.2, the BSQLM method converges quadratically to 10^{-95} after less than eight iterations. This in turn implies that the method needs few iterations to converge fully and hence taking less computational time. This is in agreement with the computational times displayed in Tables 3.1, 3.2, 3.3 and 3.4.

3.4 Conclusion

The goal of this chapter was to develop the BSQLM algorithm for solving non-similar boundary layer equations, and to evaluate its accuracy, robustness and effectiveness. To this end, we presented and used an innovative BSQLM method to solve a system of nonlinear differential equations that model magnetohydrodynamic forced convection flow adjacent to a non-isothermal wedge. The results were validated against published results, and found to be close or identical to those of Yih [42], using the Keller-box method. Moreover the accuracy of the method is independent of the size of the dependent

variables in the differential equation. Convergence analysis and grid independence tests were conducted to establish the convergence and validity of the algorithm. Computational order of convergence was proved numerically to be two, indicating that the method converges quadratically, which is in agreement with theory of quasilinearization methods by Bellman and Kalaba [71]. Furthermore, convergence was observed using few discretization points, which indicates that the BSQLM algorithm minimizes computation time. Thus the new BSQLM algorithm performs better than the Keller-box method for a class of non-similar boundary layer equations; its quadratic convergence is rapid and uses few grid points to achieve accurate results, and thus the method uses minimal computation time. Furthermore, that the accuracy of the method does not deteriorate for large values of the governing independent variables. In summary, we have presented an algorithm with rapid convergence to give accurate results in a computationally efficient manner. The success of the method can be attributed of it using the Chebyshev spectral collocation method with bivariate Lagrange interpolation both in space and time. This work contributes to the existing body of literature by providing a new and more efficient tool for solving systems of nonlinear non-similar boundary layer equations. In the next chapter, we present another new numerical method that uses Gauss-Seidel approach for solving a system of coupled nonlinear system of partial differential equations.

Chapter 4

Bivariate spectral relaxation method for unsteady three dimensional magneto hydrodynamic flow and mass transfer in a porous media

In this chapter, we propose an improved implementation of the spectral relaxation method (SRM) for solving partial differential equations. Previously, the SRM used a spectral method to discretize derivatives in space and finite differences to discretize derivatives in time [96]. In this work we apply the spectral method in both space and time. The new implementation of the SRM is tested on a system of four partial differential equations, which model an unsteady three dimensional magnetohydrodynamic flow and mass transfer in a porous media. Results are compared with previously published results of the SRM, spectral quasilinearization method (SQLM) and the Keller-box method. We have already published the work presented in this chapter, as indicated by Publication number 2 (see page *vii*).

4.1 Introduction

This work considers a system of four partial differential equations modeling an unsteady three dimensional magnetohydrodynamic flow and mass transfer in a porous media. As reported in Hayat et al. [99], examples of such flow occur in aerodynamic extrusion of plastic sheets, cooling of metallic sheets in a cooling bath and the manufacturing processes for artificial film and fibers. Due to these important

applications, many researchers have dedicated time and effort to studying such models in order to find their solutions. To find approximate solutions for equations a variety of methods are available. This particular model has been solved before by Hayat et al. [99] using the homotopy analysis method (HAM) and more recently by Motsa et al. [96] using spectral methods.

The HAM is an analytic method for approximating solutions of differential equations that was developed by Liao [100], and has since been used extensively by researchers [101–108]. With this method being analytic, accuracy and convergence are achieved by increasing the number of terms in the HAM series. Accurate results, particularly with a large embedded physical parameter multiplying the nonlinear unknown terms, may then depend on having a large number of terms. However, retaining so many terms in the solution series may be cumbersome, even with the use of symbolic computing software. The HAM also requires other arbitrarily introduced parameters, such as the convergence controlling parameter, which must be carefully selected in a separate procedure.

Another popular numerical method used by many researchers to solve unsteady boundary layer flow problems is the Keller-box method [109–113]. The Keller-box method is a finite difference-based implicit numerical scheme which was developed by Cebeci and Bradshaw [81].

Spectral based schemes have been used recently by Motsa et al. [96, 114] to solve unsteady boundary layer problems. The schemes include the spectral quasilinearization method (SQLM) and spectral relaxation method (SRM) which are accurate, easy to implement and computationally efficient. However, as observed by Motsa et al. [96], the SQLM has limitations. Firstly, it solves the coupled system directly, which may result in very large algebraic systems of equations, which in turn may require considerable computing resources. Also, developing the solution algorithm is a long process because one starts with a quasilinearization step. This contrasts with the shorter process of the SRM, in which one obtains the iteration scheme directly by requiring some terms to be evaluated at the current iteration and others at the previous iteration. The SRM employs the ideas of Gauss-Seidel relaxation to decouple a nonlinear system of PDEs into a sequence of linear PDEs, which are solved in succession. As a result the SRM is easy to implement and computationally efficient.

Both the original SRM and SQLM used in Motsa et al. [96] use finite differences to discretize derivatives in time. However, finite difference schemes converge slower than spectral methods, which is a disadvantage. This can negate the benefits of rapid convergence from using spectral collocation method to discretize in space. Furthermore, accuracy of finite difference methods requires fine grids with very small increments, with consequent increase in computation time each time the grid is refined. The equations governing this problem have been solved before by Motsa et al. [96] using the spectral relaxation method (SRM), spectral quasilinearization method (SQLM) and the Keller-box method. Motsa et al. observed

that the Keller-box method takes a significant amount of computational time; much more than either the SRM or SQLM. The Keller-box method is based entirely on finite difference schemes, whereas the SRM and SQLM use finite differences only in the time variable. For the space variable, both the SRM and SQLM use spectral methods. It is well documented from literature that spectral methods converge very quickly when the solution is smooth. This brought about the idea of using spectral methods in both space and time, in order to increase efficiency. The BSRM used here discretize both the space and time domains using spectral methods.

This chapter provides a different approach to the implementation of the SRM. Here, we use the spectral method in both space and time. We refer to the improved SRM as the bivariate spectral relaxation method (BSRM). We test the applicability of this new innovation on the PDE system mentioned above, and compare our results with those from the traditional SRM, SQLM and Keller-box as reported by Motsa et al. [96]. We are particularly concerned with comparing the computational times for the previous methods to reach the same level of accuracy as our new method.

4.2 Governing Equations

We consider the unsteady and three-dimensional flow of a viscous fluid over a stretching surface investigated by Hayat et al. [99]. The fluid is electrically conducting in the presence of a constant applied magnetic field B_0. The induced magnetic field is neglected under the assumption of a small magnetic Reynolds number. The flow is governed by the following four dimensionless partial differential equations

$$f''' + (1-\xi)\left(\frac{\eta}{2}f'' - \xi\frac{\partial f'}{\partial \xi}\right) + \xi\left[(f+g)f'' - (f')^2 - M^2 f' - \lambda f'\right] = 0, \tag{4.1}$$

$$g''' + (1-\xi)\left(\frac{\eta}{2}g'' - \xi\frac{\partial g'}{\partial \xi}\right) + \xi\left[(f+g)g'' - (g')^2 - M^2 g' - \lambda g'\right] = 0, \tag{4.2}$$

$$\theta'' + Pr(1-\xi)\left(\frac{\eta}{2}\theta' - \xi\frac{\partial \theta}{\partial \xi}\right) + Pr\xi(f+g)\theta' = 0, \tag{4.3}$$

$$\phi'' + Sc(1-\xi)\left(\frac{\eta}{2}\phi' - \xi\frac{\partial \phi}{\partial \xi}\right) + Sc\xi(f+g)\phi' - \gamma Sc\xi\phi = 0 \tag{4.4}$$

with the following boundary conditions

$$f(\xi,0) = g(\xi,0) = 0, \quad f'(\xi,0) = \theta(\xi,0) = \phi(\xi,0) = 1, \tag{4.5}$$

$$f'(\xi,\infty) = g'(\xi,\infty) = \theta(\xi,\infty) = \phi(\xi,\infty) = 0, g'(\xi,0) = c \tag{4.6}$$

In the above equations prime denotes the derivative with respect to η, c the stretching parameter is a positive constant. M is the local Hartman number, λ the local porosity parameter, Sc the Schmidt number, Pr the Prandtl number and γ the chemical reaction parameter. The initial unsteady solution can be found exactly by setting $\xi = 0$ in the above equations and solving the resulting equations. The closed form analytical solutions are given by

$$f(0,\eta) = \eta \operatorname{erfc}\left(\frac{\eta}{2}\right) + \frac{2}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right], \tag{4.7}$$

$$g(0,\eta) = c\left(\eta \operatorname{erfc}\left(\frac{\eta}{2}\right) + \frac{2}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right]\right), \tag{4.8}$$

$$\theta(0,\eta) = \operatorname{erfc}\left(\frac{\sqrt{Pr}\eta}{2}\right), \tag{4.9}$$

$$\phi(0,\eta) = \operatorname{erfc}\left(\frac{\sqrt{Sc}\eta}{2}\right). \tag{4.10}$$

Equation (4.1) - (4.4) can be expressed in the following form:

$$\begin{aligned} \Gamma_1[H_1,H_2,H_3,H_4] &= 0, \\ \Gamma_2[H_1,H_2,H_3,H_4] &= 0, \\ \Gamma_3[H_1,H_2,H_3,H_4] &= 0, \\ \Gamma_4[H_1,H_2,H_3,H_4] &= 0, \end{aligned} \tag{4.11}$$

where

$$\begin{aligned} H_1 &= \left\{f_1, \frac{\partial f_1}{\partial \eta}, \frac{\partial^2 f_1}{\partial \eta^2}, \frac{\partial^3 f_1}{\partial \eta^3}, \frac{\partial f_1}{\partial \xi}, \frac{\partial}{\partial \xi}\left(\frac{\partial f_1}{\partial \eta}\right)\right\}, \\ H_2 &= \left\{f_2, \frac{\partial f_2}{\partial \eta}, \frac{\partial^2 f_2}{\partial \eta^2}, \frac{\partial^3 f_2}{\partial \eta^3}, \frac{\partial f_2}{\partial \xi}, \frac{\partial}{\partial \xi}\left(\frac{\partial f_2}{\partial \eta}\right)\right\}, \\ H_2 &= \left\{f_3, \frac{\partial f_3}{\partial \eta}, \frac{\partial^2 f_3}{\partial \eta^2}, \frac{\partial^3 f_3}{\partial \eta^3}, \frac{\partial f_3}{\partial \xi}, \frac{\partial}{\partial \xi}\left(\frac{\partial f_3}{\partial \eta}\right)\right\}, \\ H_4 &= \left\{f_4, \frac{\partial f_4}{\partial \eta}, \frac{\partial^2 f_4}{\partial \eta^2}, \frac{\partial^3 f_4}{\partial \eta^3}, \frac{\partial f_4}{\partial \xi}, \frac{\partial}{\partial \xi}\left(\frac{\partial f_4}{\partial \eta}\right)\right\}. \end{aligned} \tag{4.12}$$

4.3 Bivariate Spectral Relaxation Method (BSRM)

In this section we introduce the Bivariate Spectral Relaxation Method (BSRM) for solving the system of nonlinear partial differential equations (4.1) - (4.4). Applying the relaxation scheme [96] to the system of

nonlinear partial differential equations gives the following linear partial differential equations;

$$f'''_{r+1} + \alpha_{1,r} f''_{r+1} + \alpha_{2,r} f'_{r+1} + \alpha_{3,r} f_{r+1} - \xi(1-\xi)\frac{\partial f'_{r+1}}{\partial \xi} = R_{1,r}, \tag{4.13}$$

$$g'''_{r+1} + \beta_{1,r} g''_{r+1} + \beta_{2,r} g'_{r+1} + \beta_{3,r} g_{r+1} - \xi(1-\xi)\frac{\partial g'_{r+1}}{\partial \xi} = R_{2,r}, \tag{4.14}$$

$$\theta''_{r+1} + \sigma_{1,r} \theta'_{r+1} + \sigma_{2,r} \theta_{r+1} - \mathrm{Pr}\xi(1-\xi)\frac{\partial \theta_{r+1}}{\partial \xi} = R_{3,r}, \tag{4.15}$$

$$\phi''_{r+1} + \omega_{1,r} \phi'_{r+1} + \omega_{2,r} \phi_{r+1} - \mathrm{Sc}\xi(1-\xi)\frac{\partial \phi_{r+1}}{\partial \xi} = R_{4,r}, \tag{4.16}$$

subject to

$$\begin{aligned} &f_{r+1}(\xi,0) = g_{r+1}(\xi,0) = 0, \quad f'_{r+1}(\xi,0) = \theta_{r+1}(\xi,0) = \phi_{r+1}(\xi,0) = 1, \\ &f'_{r+1}(\xi,\infty) = g'_{r+1}(\xi,\infty) = \theta_{r+1}(\xi,\infty) = \phi_{r+1}(\xi,\infty) = 0, g'_{r+1}(\xi,0) = c \end{aligned} \tag{4.17}$$

where the variable coefficients are given by

$$\begin{aligned} &\alpha_{1,r} = \frac{1}{2}\eta(1-\xi) + \xi g_r, \quad \alpha_{2,r} = -\xi(M^2+\lambda), \quad \alpha_{3,r} = 0, \\ &\beta_{1,r} = \frac{1}{2}\eta(1-\xi) + \xi f_r, \quad \beta_{2,r} = -\xi(M^2+\lambda), \quad \beta_{3,r} = 0, \\ &\sigma_{1,r} = \mathrm{Pr}\left(\frac{1}{2}\eta(1-\xi) + \xi(f_r+g_r)\right), \quad \sigma_{2,r} = 0 \\ &\omega_{1,r} = \mathrm{Sc}\left(\frac{1}{2}\eta(1-\xi) + \xi(f_r+g_r)\right), \quad \omega_{2,r} = -\gamma \mathrm{Sc}\,\xi \\ &R_{1,r} = \xi(f')_r^2 - \xi f_r f''_r, \quad R_{2,r} = \xi(g')_r^2 - \xi g_r g''_r, \quad R_{3,r} = 0, \quad R_{4,r} = 0 \end{aligned}$$

and r and $r+1$ denote previous and current iterations respectively. The system of linear partial differential equations (4.13) - (4.16) is discretized using the Chebyshev spectral collocation both in space (η) and time (ξ) directions. The Chebyshev collocation method is valid in the domain $[-1,1]$ in space and time. Therefore, the physical region, $\xi \in [0,1]$ is converted to the region $t \in [-1,1]$ using a linear transformation and similarly, $\eta \in [0, L_\infty]$ is converted to the region $x \in [-1,1]$. The system of linear partial differential equations (4.13) - (4.16) is decoupled. Therefore, each equation can be solved independently of the other equations in the system. We assume that the solution to equation (4.13) can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$f(\eta,\xi) \approx \sum_{p=0}^{N_\eta} \sum_{j=0}^{N_\xi} f(\eta_p,\xi_j) L_p(\eta) L_j(\xi), \tag{4.18}$$

which interpolates $f(\eta,\xi)$ at selected points in both the η and ξ directions defined by the Chebyshev-Gauss-Lobatto grid points (1.9) which ensure a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. $L_p(\eta)$ is the characteristic Lagrange cardinal polynomial as defined by equation (1.5). The number of grid points in both η and ξ directions are given by N_η and N_ξ respectively. Substituting equation (4.18) into equation (4.13) and collocating after using equations (1.12), (1.16), we get

$$\left[\mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r}\right] \mathbf{F}_{r+1,i} - \xi_i(1-\xi_i) \sum_{j=0}^{N_\xi} d_{ij}\mathbf{D}\mathbf{F}_{r+1,j} = \mathbf{R}_{1,r}, \quad i = 0,1,2,\ldots,N_\xi, \tag{4.19}$$

where $\boldsymbol{\alpha}_{v,r}$ ($v = 1,2,3$) is the diagonal matrix of the vector $[\alpha_{v,r}(\eta_0), \alpha_{v,r}(\eta_1), \cdots, \alpha_{v,r}(\eta_{N_\eta})]^T$ and $\mathbf{R}_{1,r} = [R_{1,r}(\eta_0), R_{1,r}(\eta_1), \cdots, R_{1,r}(\eta_{N_\eta})]^T$. The boundary equations are given by

$$f_{r+1,i}(\eta_{N_\eta}) = 0, \quad f'_{r+1,i}(\eta_{N_\eta}) = 1, \quad f'_{r+1,i}(\eta_0) = 0, \tag{4.20}$$

The initial unsteady solution given by equation (4.7) corresponds to $\xi = \xi_{N_\xi} = -1$. Therefore, we evaluate equation (4.19) for $i = 0,1,\cdots,N_\xi - 1$. Equation (4.19) can be expressed as

$$\left[\mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r}\right] \mathbf{F}_{r+1,i} - \xi_i(1-\xi_i) \sum_{j=0}^{N_\xi - 1} d_{ij}\mathbf{D}\mathbf{F}_{r+1,j} = \mathbf{R}_{1,i}, \quad i = 0,1,2,\ldots,N_\xi, \tag{4.21}$$

where

$$\mathbf{R}_{1,i} = \mathbf{R}_{1,r} + \xi_i(1-\xi_i) d_{iN_\xi} \mathbf{D}\mathbf{F}_{N_\xi},$$

and $\mathbf{F}_{N_\xi}$ is the known initial unsteady solution given by equation (4.7). Imposing boundary conditions for $i = 0,1,\cdots,N_\xi - 1$, equation (4.21) can be expressed as the following $N_\xi(N_\eta+1) \times N_\xi(N_\eta+1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_\xi-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_\xi-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_\xi-1,0} & A_{N_\xi-1,1} & \cdots & A_{N_\xi-1,N_\xi-1} \end{bmatrix} \begin{bmatrix} \mathbf{F}_0 \\ \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_{N_\xi-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{1,0} \\ \mathbf{R}_{1,1} \\ \vdots \\ \mathbf{R}_{1,N_\xi-1} \end{bmatrix}, \tag{4.22}$$

where

$$A_{i,i} = \mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r} - \xi_i(1-\xi_i)d_{ii}\mathbf{D} \tag{4.23}$$

$$A_{i,j} = -\xi_i(1-\xi_i)d_{ij}\mathbf{D}, \quad \text{when } i \neq j, \tag{4.24}$$

Imposing initial boundary conditions and applying the Chebyshev bivariate collocation as described above on equations, (4.14), (4.15) and (4.16) we get

$$\left[\mathbf{D}^3 + \boldsymbol{\beta}_{1,r}\mathbf{D}^2 + \boldsymbol{\beta}_{2,r}\mathbf{D} + \boldsymbol{\beta}_{3,r}\right]\mathbf{G}_{r+1,i} - \xi_i(1-\xi_i)\sum_{j=0}^{N_\xi-1} d_{ij}\mathbf{D}\mathbf{G}_{r+1,j} = \mathbf{R}_{2,i}, \tag{4.25}$$

$$\left[\mathbf{D}^2 + \boldsymbol{\sigma}_{1,r}\mathbf{D} + \boldsymbol{\sigma}_{2,r}\right]\boldsymbol{\Theta}_{r+1,i} - \mathrm{Pr}\xi_i(1-\xi_i)\sum_{j=0}^{N_\xi-1} d_{ij}\boldsymbol{\Theta}_{r+1,j} = \mathbf{R}_{3,i}, \tag{4.26}$$

$$\left[\mathbf{D}^2 + \boldsymbol{\omega}_{1,r}\mathbf{D} + \boldsymbol{\omega}_{2,r}\right]\boldsymbol{\Phi}_{r+1,i} - \mathrm{Sc}\xi_i(1-\xi_i)\sum_{j=0}^{N_\xi-1} d_{ij}\boldsymbol{\Phi}_{r+1,j} = \mathbf{R}_{4,i}, \tag{4.27}$$

where

$$\mathbf{R}_{2,i} = \mathbf{R}_{2,r} + \xi_i(1-\xi_i)d_{iN_\xi}\mathbf{D}\mathbf{G}_{N_\xi}, \tag{4.28}$$

$$\mathbf{R}_{3,i} = \mathbf{R}_{3,r} + \mathrm{Pr}\xi_i(1-\xi_i)d_{iN_\xi}\boldsymbol{\Theta}_{N_\xi}, \tag{4.29}$$

$$\mathbf{R}_{4,i} = \mathbf{R}_{4,r} + \mathrm{Sc}\xi_i(1-\xi_i)d_{iN_\xi}\boldsymbol{\Phi}_{N_\xi}, \tag{4.30}$$

and the vectors $\mathbf{G}_{N_\xi}$, $\boldsymbol{\Theta}_{N_\xi}$ and $\boldsymbol{\Phi}_{N_\xi}$ are the known initial unsteady solutions given by equations (4.8), (4.9) and (4.10) respectively. Imposing boundary conditions for $i = 0, 1, \cdots, N_\xi - 1$, equations (4.25), (4.26) and (4.27) can be expressed as the following $N_\xi(N_\eta+1) \times N_\xi(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,N_\xi-1} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,N_\xi-1} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\xi-1,0} & B_{N_\xi-1,1} & \cdots & B_{N_\xi-1,N_\xi-1} \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_{N_\xi-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{2,0} \\ \mathbf{R}_{2,1} \\ \vdots \\ \mathbf{R}_{2,N_\xi-1} \end{bmatrix}, \tag{4.31}$$

$$\begin{bmatrix} C_{0,0} & C_{0,1} & \cdots & C_{0,N_\xi-1} \\ C_{1,0} & C_{1,1} & \cdots & C_{1,N_\xi-1} \\ \vdots & \vdots & \ddots & \vdots \\ C_{N_\xi-1,0} & C_{N_\xi-1,1} & \cdots & C_{N_\xi-1,N_\xi-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Theta}_0 \\ \boldsymbol{\Theta}_1 \\ \vdots \\ \boldsymbol{\Theta}_{N_\xi-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3,0} \\ \mathbf{R}_{3,1} \\ \vdots \\ \mathbf{R}_{3,N_\xi-1} \end{bmatrix}, \tag{4.32}$$

$$\begin{bmatrix} E_{0,0} & E_{0,1} & \cdots & E_{0,N_\xi-1} \\ E_{1,0} & E_{1,1} & \cdots & E_{1,N_\xi-1} \\ \vdots & \vdots & \ddots & \vdots \\ E_{N_\xi-1,0} & E_{N_\xi-1,1} & \cdots & E_{N_\xi-1,N_\xi-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Phi}_0 \\ \boldsymbol{\Phi}_1 \\ \vdots \\ \boldsymbol{\Phi}_{N_\xi-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{4,0} \\ \mathbf{R}_{4,1} \\ \vdots \\ \mathbf{R}_{4,N_\xi-1} \end{bmatrix}, \tag{4.33}$$

where

$$B_{i,i} = \mathbf{D}^3 + \boldsymbol{\beta}_{1,r}\mathbf{D}^2 + \boldsymbol{\beta}_{2,r}\mathbf{D} + \boldsymbol{\beta}_{3,r} - \xi_i(1-\xi_i)d_{ii}\mathbf{D} \tag{4.34}$$

$$B_{i,j} = -\xi_i(1-\xi_i)d_{ij}\mathbf{D}, \quad \text{when } i \neq j, \tag{4.35}$$

$$C_{i,i} = \mathbf{D}^2 + \boldsymbol{\sigma}_{1,r}\mathbf{D} + \boldsymbol{\sigma}_{2,r} - \mathrm{Pr}\xi_i(1-\xi_i)d_{ii}\mathbf{I} \tag{4.36}$$

$$C_{i,j} = -\mathrm{Pr}\xi_i(1-\xi_i)d_{ij}\mathbf{I}, \quad \text{when } i \neq j, \tag{4.37}$$

$$E_{i,i} = \mathbf{D}^2 + \boldsymbol{\omega}_{1,r}\mathbf{D} + \boldsymbol{\omega}_{2,r} - \mathrm{Sc}\xi_i(1-\xi_i)d_{ii}\mathbf{I} \tag{4.38}$$

$$E_{i,j} = -\mathrm{Sc}\xi_i(1-\xi_i)d_{ij}\mathbf{I}, \quad \text{when } i \neq j, \tag{4.39}$$

and $\mathbf{I}$ is the standard $(N_\eta+1)\times(N_\eta+1)$ identity matrix. We obtain the numerical solutions for $g(\eta,\xi)$, $\theta(\eta,\xi)$ and $\phi(\eta,\xi)$ by solving matrix equations (4.31), (4.32) and (4.33) iteratively for $r = 1,2,\cdots M$, where M is the number of iterations to be used. Equations (4.8), (4.9) and (4.10) are used as initial guesses.

4.4 Results and Discussion

In this section we present the numerical solutions of the three dimensional unsteady three dimensional magneto-hydrodynamic flow and mass transfer in a porous media obtained using the BSQLM algorithm. In our computations the the η domain was truncated to $\eta_\infty = 20$. This value gave accurate results for all the quantities of physical interest. To get accurate solutions, $N_\eta = 60$ collocation points were used to discretize the space variable η and only $N_\xi = 10$ collocation points were enough for the time variable ξ. The results are validated by using the residual error. The residual error after r iterations over all

$i = 0, 1, 2, \ldots, N_\xi$ is given by

$$||Res(f_k)||_\infty = ||\Gamma_k \left[H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, H_4^{(i)} \right] ||_\infty \tag{4.40}$$

for $k = 1, 2, 3, 4$.

Table 4.1 Values of $f''(0,\xi)$, $g''(0,\xi)$, $\theta'(0,\xi)$ and $\phi'(0,\xi)$ when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

ξ	BSRM ($N_\xi = 10$)	SRM ($N_\xi = 2000$)	SQLM ($N_\xi = 2000$)	Keller-box ($N_\xi = 2000$)
		$f''(0,\xi)$		
0.1	-0.851257	-0.851257	-0.851257	-0.851257
0.3	-1.316705	-1.316705	-1.316705	-1.316705
0.5	-1.685306	-1.685306	-1.685306	-1.685306
0.7	-1.992608	-1.992608	-1.992608	-1.992608
0.9	-2.259335	-2.259335	-2.259335	-2.259335
		$g''(0,\xi)$		
0.1	-0.417150	-0.417150	-0.417150	-0.417150
0.3	-0.639602	-0.639602	-0.639602	-0.639602
0.5	-0.817649	-0.817649	-0.817649	-0.817649
0.7	-0.966603	-0.966603	-0.966603	-0.966603
0.9	-1.095983	-1.095983	-1.095983	-1.095983
		$\theta'(0,\xi)$		
0.1	-0.710882	-0.710882	-0.710882	-0.710882
0.3	-0.742842	-0.742842	-0.742842	-0.742842
0.5	-0.765244	-0.765244	-0.765244	-0.765244
0.7	-0.777270	-0.777270	-0.777270	-0.777270
0.9	-0.770807	-0.770807	-0.770807	-0.770807
		$\phi'(0,\xi)$		
0.1	-0.634443	-0.634443	-0.634443	-0.634443
0.3	-0.766867	-0.766867	-0.766867	-0.766867
0.5	-0.891207	-0.891207	-0.891207	-0.891207
0.7	-1.010045	-1.010045	-1.010045	-1.010045
0.9	-1.125549	-1.125549	-1.125549	-1.125549
CPU time	0.47	18.90	83.24	900.30

As earlier mentioned, this problem has been solved before by Motsa et. al. [96] using the spectral relaxation method (SRM), spectral quasilinearization method (SQLM) and the Keller-box method. The results from their paper combined with the present results of the BSRM are shown in Table 4.1. It can be observed from the table that the Keller-box method takes a significant amount of computational time than the SRM and SQLM. From the results shown in the table it is evident that the BSRM is by far superior

than the other methods in terms of computational time taken to reach the same level of accuracy. In Table 4.1, we also show the number of grid points required by each of the methods to discretize in time. All the finite difference based discretization required 2000 grid points compared to the spectral discretization of the BSRM which required only 10 grid points to reach the same level of accuracy. We can infer that the BSRM is better than finite differences coupled SRM in terms of computational speed and accuracy and the better accuracy could be the result of applying spectral collocation with uniform accuracy level in both η and ξ directions.

Table 4.2 The residual errors and convergence rates of f when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

	$\|Res(\mathbf{f})\|_\infty$			Convergence Rates		
Iter.	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$
1	2.14×10^{-2}	2.36×10^{-1}	3.72×10^{-1}	1.14	1.00	0.98
2	6.03×10^{-4}	1.43×10^{-2}	2.24×10^{-2}	0.50	1.01	1.01
3	1.05×10^{-5}	8.58×10^{-4}	1.43×10^{-3}	1.53	1.00	1.00
4	1.36×10^{-6}	5.04×10^{-5}	8.85×10^{-5}	1.06	1.01	1.00
5	5.95×10^{-8}	2.98×10^{-6}	5.52×10^{-6}	0.97	1.02	1.00
6	2.18×10^{-9}	1.70×10^{-7}	3.40×10^{-7}	0.99	1.00	1.00
7	8.84×10^{-11}	9.06×10^{-9}	2.07×10^{-8}	0.95	1.00	1.00
8	3.75×10^{-12}	4.85×10^{-10}	1.27×10^{-9}	0.85	0.99	1.00

In Table 4.2, the residual errors and convergence rates of f when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1$ are displayed. We observe that the convergence rate is linear and the residual error is smaller near $\xi = 0$. We observe that increasing the number of iterations, decreases the residual error and hence increasing the accuracy of the method. This basically implies that results obtained after couple of iterations are more accurate than results obtained just after few iterations. Increasing the iterations does not have a direct impact on the order of convergence of the method as expected. However, we note that an increase in ξ results in an increase in the residual error. The increase is insignificant as far as accuracy of the method is concerned. The same results are observed in Table 4.3, where the same parameters were used to generate the results.

Table 4.3 The residual errors and convergence rates of g when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

	$\|Res(\mathbf{g})\|_\infty$			Convergence Rates		
Iter.	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$
1	4.17×10^{-3}	4.93×10^{-2}	7.83×10^{-2}	0.94	1.01	0.99
2	4.96×10^{-5}	1.40×10^{-3}	2.21×10^{-3}	0.75	1.03	1.04
3	7.66×10^{-7}	3.81×10^{-5}	6.39×10^{-5}	1.26	1.13	1.18
4	3.38×10^{-8}	9.21×10^{-7}	1.58×10^{-6}	1.07	0.76	0.56
5	6.56×10^{-10}	1.38×10^{-8}	2.01×10^{-8}	0.83	0.68	0.92
6	9.49×10^{-12}	5.75×10^{-10}	1.78×10^{-9}	1.00	1.27	1.18
7	2.82×10^{-13}	6.58×10^{-11}	1.93×10^{-10}	0.89	1.08	1.04
8	8.39×10^{-15}	4.19×10^{-12}	1.39×10^{-11}	0.95	1.00	1.00

The residual error graphs of equations (4.1) - (4.4) are presented in Fig. 4.1 - Fig. 4.4 respectively. In Fig. 4.1 and Fig. 4.2, we observe that the residual error is reduced with an increase in the iterations of the scheme. The rate of reduction of the residual error appears to be linear. The residual error is minimum at $\xi = 0$ and is increased sharply near 0 until a certain level is reached after which it is almost constant. The residual error appears to be nearly uniform in $0 < \xi \leq 1$ or increases only slightly. It is also observed that the order of magnitude of the residual error can be seen to be small in the $0 \leq \xi \leq 1$ interval. Lastly, after only two iterations the residual error appears to be less than 0.01 in the entire range of ξ. The small residual error using only a few iteration points points to the accuracy of the method. This error can be decreased at a linear rate with an increase in the number of iterations. The decrease in the error with additional iterations suggests that the iteration scheme converges. It should be noted that when $\xi = 0$, governing equations reduce to a linear system that can be solved directly using the spectral collocation method with discretization only in η without the use of relaxation and iterations. This explains why the best accuracy is observed at $\xi = 0$. The near uniformity of the residual error in $0 < \xi \leq 1$ can be attributed to the use of Lagrange polynomial basis functions whose error is known to be uniformly distributed in the interpolating region. We can therefore conclude that the method gives accurate results, the rate of convergence of the method is linear and that the method requires only a few iterations to give very accurate results.

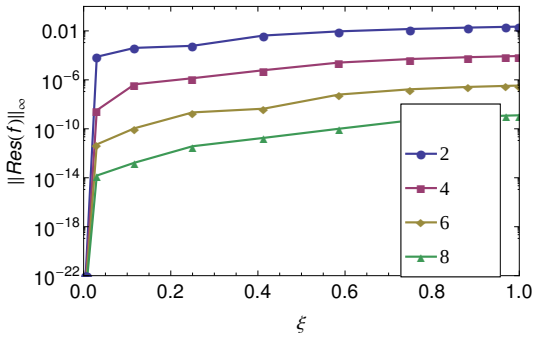

Fig. 4.1 Residual graph of $f(\eta,\xi)$

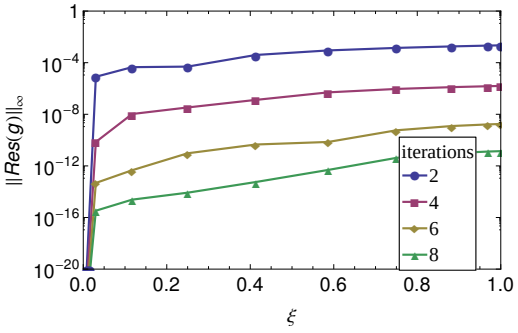

Fig. 4.2 Residual graph of $g(\eta,\xi)$

We observe that the residual error for the first iteration appears to be very small in Fig. 4.3 and Fig. 4.4. The residual error is the same for all iterations greater than one. The residual error increases with an increase in ξ. We also observe that the residual error is smaller than the one for f and g even when fewer iterations are used. The observation that the residual error for the momentum and energy is small even for the 1st iteration is perhaps the most interesting finding of the study. This means that when using the proposed approach, the best possible results that can be achieved by the method can be obtained after using just one iteration. Further increase in the number of iterations doesn't improve the accuracy of the solution. After one iteration of the momentum equations for f and g the energy and mass transfer equations reduce to linear homogeneous equations whose solution appears to be marginally influenced by variations in f_r and g_r for $r > 1$. Since with just one iteration we obtain extremely accurate results for θ and ϕ, the implication is that in solving for energy and momentum equations for such a problem, it is not necessary to iterate. It is enough to just use the initial approximation. Is is worth noting that the energy and mass transfer equations are homogeneous equations in θ and ϕ respectively. It is possible that the findings obtained in this study are only applicable in such equations. This has to be investigated further.

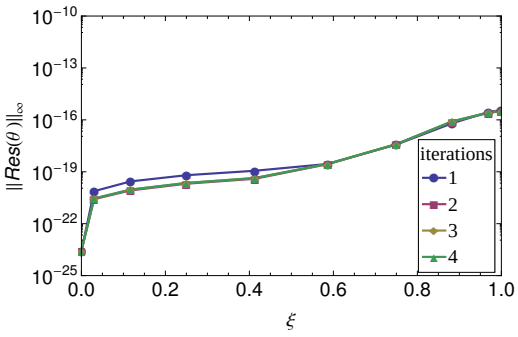

Fig. 4.3 Residual graph of $\theta(\eta,\xi)$

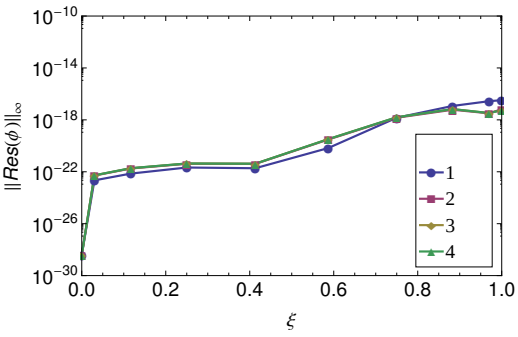

Fig. 4.4 Residual graph of $\phi(\eta,\xi)$

The convergence graphs of equations (4.1) - (4.4) are presented in Fig. 4.5 - Fig. 4.6 respectively. In 4.5 - Fig. 4.6, the residual error decreases linearly with an increase in the number of iterations.

The residual error is smallest when ξ is near zero and largest when ξ is large. This is seen from the convergence level which is near 10^{-20} for $\xi = 0.25$ and about 10^{-15} for $\xi = 0.75$. The slope of the residual error graphs is the same for all values of ξ. Full convergence is achieved after 13 iterations for both $\xi = 0.75$ and $\xi = 1$. For $\xi = 0.25$ full convergence is achieved after 16 iterations but at a much smaller magnitude of residual error. The decrease in the residual error with increase in iterations suggests that the iteration scheme converges. Small residual error near zero suggests that best accuracy (after full convergence) is observed near zero. The method converges (fewer iterations needed to attain full convergence) at or near $\xi = 1$. However, the convergence efficiency doesn't translate to better accuracy because, as can be seen for the case of ξ values near zero, the convergence level is 10^{-16}. The same slope for all the graphs means that the convergence rates of the method is the same for all values of ξ. The method is convergent and very accurate in whole time domain $\xi \in [0,1]$ which translates to $\tau \in [0,\infty)$. The method converges with nearly the same convergence rate for all values of ξ. The method gives the best accuracy near $\xi = 0$ and less accurate, comparatively, at or near $\xi = 1$. We note that even at $\xi = 1$, the method gives very accurate results with a residual error norm of about 10^{-15}. This is one of the highlights of this investigation.

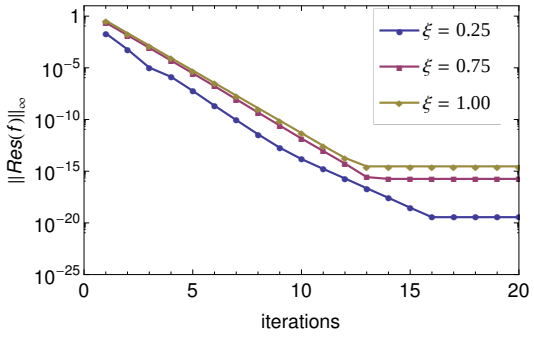

Fig. 4.5 Convergence graph of $f(\eta,\xi)$

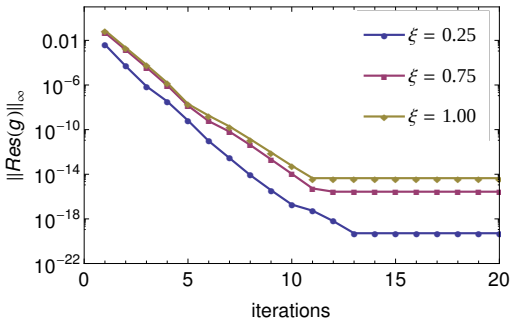

Fig. 4.6 Convergence graph of $g(\eta,\xi)$

4.5 Conclusion

The aim of this chapter was to introduce the bivariate spectral relaxation method (BSRM) for solving systems of coupled partial differential equations (PDEs). We derived the algorithm for the BSRM by extending a previous method that used finite differences (SRM) to now incorporate bivariate interpolation. We used the method to solve the system of equations that model an unsteady three dimensional magnetohydrodynamic flow and mass transfer in a porous media. We compared the performance of the new BSRM in solving these equations with data reported for the original SRM, SQLM and Keller-box method. The new approach offers spectral accuracy in both variables. Furthermore, this accuracy is achieved

with many fewer grid points than are needed for finite difference methods and so there is also improved efficiency and that the residual errors rapidly approach zero. The method has been shown to converge linearly with a convergence rate of one.

The results for the energy and mass transfer, given approximate solutions for θ and ϕ, respectively, are not dependent on the successive approximations for either f or g. It is enough to use a single iteration of f or g to give the most accurate result for θ. The method also gives accurate results in the complete time domain $\xi \in [0,1]$ ($\tau \in [0,\infty)$). We also found that the new method performs significantly better than the previous SRM, SQLM and Keller-box method in terms of computational time, due to the application of spectral collocation in both directions. The algorithm for the BSRM is straightforward because it involves using only known formulae for discretization in Chebyshev spectral collocation. The method is therefore recommended as a viable option for solving unsteady boundary layer flows over an impulsively stretched sheet because of the demonstrated computational benefits and the effortless derivation of the BSRM. The method could also be applied to other two dimensional problems or even be extended to three dimensional coupled non-linear systems, and its performance with these evaluated.

In the next chapter, we introduce another new approach for solving a system of n nonlinear partial differential equations, which we call the bivariate spectral local linearization method. It will be compared with the previously published bivariate spectral quasilinearization method and the bivariate spectral relaxation method from this chapter for solving systems of nonlinear partial differential equations.

Chapter 5

A comparison of bivariate pseudo spectral methods for nonlinear systems of partial differential equations

In this chapter, three pseudospectral methods are analyzed and compared. The methods are the bivariate spectral quasilinearization method (BSQLM, see Chapters 2 and 3), the bivariate spectral relaxation method (BSRM, see Chapter 4), and we introduce the bivariate spectral local linearization method (BSLLM). General overall performance is discussed in detail. A system of three nonlinear coupled partial differential equations that model an unsteady three-dimensional MHD-boundary-layer flow due to an impulsive motion of a stretching surface is used to demonstrate the accuracy, computational time and general performance of the three pseudospectral methods. Results of the comparison are presented in graphs and tables. In addition, the BSQLM and BSLLM are described for any general system of n equations. This generalization is done for the first time this thesis.

5.1 Introduction

Pseudospectral methods have been developed to solve ordinary differential equations [115–118] and partial differential equations [119–122] that model various scientific phenomena. There are different ways to implement pseudospectral methods, namely through spectral relaxation, spectral local linearization and spectral quasilinearization. The spectral relaxation method is a first order accurate pseudospectral method that has been applied to solve ordinary differential equations arising in fluid mechanic [123–126]. It was extended to solve partial differential equations [127–130, 96] by the space and time derivatives

being discretized using spectral collocation and finite differences respectively. Finite differences require a fine mesh (hence many grid points) for convergence while spectral methods need only a coarse mesh (hence few grid points). Thus combining finite differences with spectral methods compromises the computational efficiency of the spectral method. As shown in the previous chapter, this method can be extended with spectral methods applied independently on both space and time derivatives to solve systems of nonlinear partial differential equations [131, 132]. This development, named the bivariate spectral relaxation method (BSRM), decouples and linearize systems of nonlinear coupled partial differential equations. The resulting linear decoupled system of equations is then solved using a spectral collocation method.

Other methods have been developed to solve systems of nonlinear partial differential equations. One of these, the bivariate spectral local linearization method (BSLLM) has been used to solve systems of non-linear partial differential equations [133, 134]. It is a first order accurate method that decouples the system of partial differential equations. Nonlinear partial differential equations are then linearized using the quasilinearization technique developed by Bellman and Kalaba [71]. The algorithm was generalized in by Motsa [134] for solving a system of three equations. In this chapter, we extend the generalization of Motsa's the method in [137] method to a system of n nonlinear partial differential equations.

Another pseudospectral method, the bivariate spectral quasilinearization method (BSQLM), was first developed to solve a nonlinear evolution partial differential equation, as described in Chapters 2 and 3 [135]. It has since been extended to other systems of nonlinear partial differential equations [136, 137]. The BSQLM is a second order accurate method, wherein the nonlinear PDEs are linearized using the quasilinearization technique [71]. The linearized PDEs are then solved using the spectral collocation approach. In this chapter, the BSQLM is generalized to a system of n nonlinear PDEs.

After generalizing the BSQLM and BSLLM to solve a system of n nonlinear system of PDEs, the main aim of this chapter is to compare the general overall performance of these two bivariate pseudospectral methods with the BSRM.

For numerical experiments in this chapter, a system of three nonlinear coupled partial differential equations that model the flow of an unsteady three-dimensional MHD-boundary-layer due to the impulsive motion of a stretching surface is solved using the three pseudospectral methods. In comparing the accuracy and general performance of the three pseudospectral methods, graphs are presented in the results section, together with tables showing the physical quantities of interest in fluid mechanics and computational times.

The rest remainder of the this chapter is arranged as follows. In Section 5.2, the BSQLM and BSLLM methods are presented in a generalized form. These methods together with the BSRM are then implemented in Section 5.3. The results are discussed in Section 5.4 with conclusions in Section 5.5.

5.2 Pseudospectral Numerical Methods

In this section, we present the general form of the pseudospectral numerical methods, namely, the bivariate spectral quasilinearisation method (BSQLM) and bivariate spectral local linearisation method (BSLLM). We briefly discuss the bivariate spectral relaxation method (BSRM). The methods are discussed in detail on how to use them to solve systems of n nonlinear partial differential equations.

Bivariate Spectral Quasilinearisation Method

In this section we introduce the Bivariate Spectral Quasilinearization Method (BSQLM) for approximating solutions of system of nonlinear partial differential equations. We develop the bivariate spectral quasilinearization method for a general system of n nonlinear partial differential equations. We consider system of n nonlinear partial differential equations of the form,

$$\begin{aligned} \Gamma_1[H_1,H_2,\ldots,H_n] &= 0,\\ \Gamma_2[H_1,H_2,\ldots,H_n] &= 0,\\ &\vdots\\ \Gamma_n[H_1,H_2,\ldots,H_n] &= 0, \end{aligned} \tag{5.1}$$

where

$$\begin{aligned} H_1 &= \left\{ f_1, \frac{\partial f_1}{\partial \eta}, \frac{\partial^2 f_1}{\partial \eta^2}, \ldots, \frac{\partial^p f_1}{\partial \eta^p}, \frac{\partial f_1}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_1}{\partial \eta}\right) \right\},\\ H_2 &= \left\{ f_2, \frac{\partial f_2}{\partial \eta}, \frac{\partial^2 f_2}{\partial \eta^2}, \ldots, \frac{\partial^p f_2}{\partial \eta^p}, \frac{\partial f_2}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_2}{\partial \eta}\right) \right\},\\ &\vdots\\ H_n &= \left\{ f_n, \frac{\partial f_n}{\partial \eta}, \frac{\partial^2 f_n}{\partial \eta^2}, \ldots, \frac{\partial^p f_n}{\partial \eta^p}, \frac{\partial f_n}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_n}{\partial \eta}\right) \right\}. \end{aligned} \tag{5.2}$$

The order of differentiation is denoted by p, the solution by $f_k(\eta,\zeta)$ for $k = 1,2,\ldots,n$ and Γ_k for $k = 1,2,\ldots,n$ are non-linear operators containing all the spatial derivatives and time derivatives of $f_k(\eta,\zeta)$. We assume that the solution can be approximated by a bivariate Lagrange interpolation

polynomial of the form

$$f_k(\eta,\zeta) \approx \sum_{i=0}^{N_\eta} \sum_{j=0}^{N_\zeta} f_k(\eta_i,\zeta_j)\mathscr{L}_i(\eta)\mathscr{L}_j(\zeta), \tag{5.3}$$

for $k = 1,2,\ldots,n$. The bivariate Lagrange interpolation polynomial interpolates $f_k(\eta,\zeta)$ at selected grid points (η_i,ζ_j) in both the η and ζ directions, for $i = 0,1,2,\ldots,N_\eta$ and $j = 0,1,2,\ldots,N_\zeta$. These grid points are given by equation (1.9). The nonlinear operators Γ_k, for $k = 1,2,3,\ldots,n$ are first linearized using the quasilinearisation technique as defined by Bellman and Kalaba [71]. The quasilinearisation method is based on the Taylor series expansion of Γ_k about some previous iteration. We assume that the difference between the previous and current solution and all it's derivatives are small. Applying the quasilinearisation method yields the following

$$\begin{aligned}\Gamma_k\left[H_1,H_2,\ldots,H_n\right] \approx{}& \left(H_{1,r+1}-H_{1,r},H_{2,r+1}-H_{2,r},\ldots,H_{n,r+1}-H_{n,r}\right)\cdot\nabla\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right]\\ &+\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right],\end{aligned} \tag{5.4}$$

where r and $r+1$ denote previous and current iterations respectively and ∇ is a vector of the partial derivatives which is defined as

$$\nabla = \left\{\nabla_{f_1},\nabla_{f_2},\ldots,\nabla_{f_n}\right\}. \tag{5.5}$$

We define

$$\begin{aligned}
\nabla_{f_1} &= \left\{\frac{\partial}{\partial f_1},\frac{\partial}{\partial f_1'},\frac{\partial}{\partial f_1''},\ldots,\frac{\partial}{\partial f_1^{(p)}},\frac{\partial}{\partial\left(\frac{\partial f_1}{\partial\zeta}\right)},\frac{\partial}{\partial\left(\frac{\partial f_1'}{\partial\zeta}\right)}\right\},\\
\nabla_{f_2} &= \left\{\frac{\partial}{\partial f_2},\frac{\partial}{\partial f_2'},\frac{\partial}{\partial f_2''},\ldots,\frac{\partial}{\partial f_2^{(p)}},\frac{\partial}{\partial\left(\frac{\partial f_2}{\partial\zeta}\right)},\frac{\partial}{\partial\left(\frac{\partial f_2'}{\partial\zeta}\right)}\right\},\\
&\vdots\\
\nabla_{f_n} &= \left\{\frac{\partial}{\partial f_n},\frac{\partial}{\partial f_n'},\frac{\partial}{\partial f_n''},\ldots,\frac{\partial}{\partial f_n^{(p)}},\frac{\partial}{\partial\left(\frac{\partial f_n}{\partial\zeta}\right)},\frac{\partial}{\partial\left(\frac{\partial f_n'}{\partial\zeta}\right)}\right\},
\end{aligned} \tag{5.6}$$

where the prime denotes differentiation with respect to η. The linearized equation (5.4) can be expressed in a compact form as

$$\begin{aligned}\sum_{s=1}^{n} H_{s,r+1}\cdot\nabla_{f_s}\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right] = {}& \sum_{s=1}^{n} H_{s,r}\cdot\nabla_{f_s}\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right]\\ &-\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right]\end{aligned} \tag{5.7}$$

for $k = 1,2,\ldots,n$. Equation (5.7) forms a system of n coupled linear partial differential equations. They are solved iteratively for $f_1(\eta,\zeta), f_2(\eta,\zeta),\ldots,f_n(\eta,\zeta)$. Equation (5.7) can further be expressed as follows:

$$\begin{aligned}
&\sum_{v=1}^{n}\left[\sum_{s=0}^{p}\alpha_{v,s,r}^{(1)}(\eta,\zeta)f_{v,r+1}^{(s)}+\beta_{v,r}^{(1)}(\eta,\zeta)\frac{\partial f_{v,r+1}}{\partial\zeta}+\gamma_{v,r}^{(1)}(\eta,\zeta)\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r+1}}{\partial\eta}\right)\right]=R_1(\eta,\zeta),\\
&\sum_{v=1}^{n}\left[\sum_{s=0}^{p}\alpha_{v,s,r}^{(2)}(\eta,\zeta)f_{v,r+1}^{(s)}+\beta_{v,r}^{(2)}(\eta,\zeta)\frac{\partial f_{v,r+1}}{\partial\zeta}+\gamma_{v,r}^{(2)}(\eta,\zeta)\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r+1}}{\partial\eta}\right)\right]=R_2(\eta,\zeta),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&\sum_{v=1}^{n}\left[\sum_{s=0}^{p}\alpha_{v,s,r}^{(n)}(\eta,\zeta)f_{v,r+1}^{(s)}+\beta_{v,r}^{(n)}(\eta,\zeta)\frac{\partial f_{v,r+1}}{\partial\zeta}+\gamma_{v,r}^{(n)}(\eta,\zeta)\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r+1}}{\partial\eta}\right)\right]=R_n(\eta,\zeta),
\end{aligned}\tag{5.8}$$

where $\alpha_{n,p,r}^{(k)}(\eta,\zeta)$, $\beta_{v,r}^{(k)}(\eta,\zeta)$ and $\gamma_{v,r}^{(k)}(\eta,\zeta)$ are variable coefficients of $f_{n,r+1}^{(p)}$, $\frac{\partial f_{v,r+1}}{\partial\zeta}$, and $\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r+1}}{\partial\eta}\right)$, respectively. These variable coefficients correspond to the k^{th} equation, for $k = 1,2,\ldots,n$. The constant p denotes the order of differentiation. Thus, we have

$$\alpha_{n,p,r}^{(k)}(\eta,\zeta)=\frac{\partial\Gamma_k}{\partial f_{n,r}^{(p)}},\qquad \beta_{v,r}^{(k)}(\eta,\zeta)=\frac{\partial\Gamma_k}{\partial\left(\frac{\partial f_{v,r}}{\partial\zeta}\right)},\qquad \gamma_{v,r}^{(k)}(\eta,\zeta)=\frac{\partial\Gamma_k}{\partial\left(\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r}}{\partial\eta}\right)\right)}.\tag{5.9}$$

The right hand side for the kth equation is given by

$$\begin{aligned}
R_k(\eta,\zeta)=&\sum_{v=1}^{n}\left[\sum_{s=0}^{p}\alpha_{v,s,r}^{(k)}(\eta,\zeta)f_{v,r}^{(s)}+\beta_{v,r}^{(k)}(\eta,\zeta)\frac{\partial f_{v,r}}{\partial\zeta}+\gamma_{v,r}^{(k)}(\eta,\zeta)\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r}}{\partial\eta}\right)\right]\\
&-\Gamma_k\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right].
\end{aligned}\tag{5.10}$$

Equation (5.8) is evaluated at the Chebyshev-Gauss-Lobbatto grid points ζ_j ($j = 0,1,\ldots,N_\zeta$) and η_i ($i = 0,1,\ldots,N_\eta$). The values of the ζ derivatives are computed at the Chebyshev-Gauss-Lobatto grid points (η_i,ζ_j), as (for $j = 0,1,2,\ldots,N_\zeta$)

$$\begin{aligned}
\left.\frac{\partial f_n}{\partial\zeta}\right|_{(\eta_i,\zeta_j)}&=\sum_{\omega=0}^{N_\eta}\sum_{\mu=0}^{N_\zeta}f_n(\eta_\omega,\zeta_\mu)\mathscr{L}_\omega(\eta_i)\frac{d\mathscr{L}_\mu(\zeta_j)}{d\zeta}\\
&=\sum_{\mu=0}^{N_\zeta}d_{j\mu}f_n(\eta_i,\zeta_\mu)
\end{aligned}\tag{5.11}$$

where $d_{j\mu}=\frac{d\mathscr{L}_\mu(\zeta_j)}{d\zeta}$ is the jth and μth entry of the standard first derivative Chebyshev differentiation matrix of size $(N_\zeta+1)\times(N_\zeta+1)$, given by [138, 41]. The values of the space derivatives at the

Chebyshev-Gauss-Lobatto points (η_i, ζ_j) (for $i = 0, 1, 2, \ldots, N_\eta$) are similarly computed as

$$\begin{aligned}\left.\frac{\partial f_n}{\partial \eta}\right|_{(\eta_i,\zeta_j)} &= \sum_{\omega=0}^{N_\eta}\sum_{\mu=0}^{N_\zeta} f_n(\eta_\omega,\zeta_\mu)\frac{d\mathscr{L}_\omega(\eta_i)}{d\eta}\mathscr{L}_\mu(\zeta_j) \\ &= \sum_{\omega=0}^{N_\eta} D_{i\omega} f_n(\eta_\omega,\zeta_j),\end{aligned} \tag{5.12}$$

where $D_{i\omega} = \frac{d\mathscr{L}_\omega(\eta_i)}{d\eta}$, is the ith and ωth entry of the standard first derivative Chebyshev differentiation matrix of size $(N_\eta + 1) \times (N_\eta + 1)$ as defined in [138, 41]. Higher, pth order derivatives are defined as

$$\left.\frac{\partial^p f_n}{\partial \eta^p}\right|_{(\eta_i,\zeta_j)} = \sum_{\omega=0}^{N_\eta} D^p_{i\omega} f_n(\eta_\omega,\zeta_j) = \mathbf{D}^p \mathbf{F}_{n,j}, \quad i = 0, 1, 2, \ldots, N_\eta, \tag{5.13}$$

where the vector $\mathbf{F}_{n,j}$ is defined as

$$\mathbf{F}_{n,j} = [f_n(\eta_0,\zeta_j), f_n(\eta_1,\zeta_j), \ldots, f_n(\eta_{N_\eta},\zeta_j)]^T. \tag{5.14}$$

and the superscript T denotes matrix transpose. Substituting equations (5.11), (5.12), (5.13) into equation (5.8), yields

$$\begin{aligned}
\sum_{v=1}^{n}\left[A^{(i)}_{1,v}\mathbf{F}_{v,i} + \boldsymbol{\beta}^{(1)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{v,j} + \boldsymbol{\gamma}^{(1)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{v,j}\right] &= \mathbf{R}_{1,i}, \\
\sum_{v=1}^{n}\left[A^{(i)}_{2,v}\mathbf{F}_{v,i} + \boldsymbol{\beta}^{(2)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{v,j} + \boldsymbol{\gamma}^{(2)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{v,j}\right] &= \mathbf{R}_{2,i}, \\
&\vdots \\
\sum_{v=1}^{n}\left[A^{(i)}_{n,v}\mathbf{F}_{v,i} + \boldsymbol{\beta}^{(n)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{v,j} + \boldsymbol{\gamma}^{(n)}_{v,r}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{v,j}\right] &= \mathbf{R}_{n,i},
\end{aligned} \tag{5.15}$$

where

$$A^{(i)}_{1,1} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(1)}_{1,s,r}\mathbf{D}^{(s)}, \quad A^{(i)}_{1,2} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(1)}_{2,s,r}\mathbf{D}^{(s)}, \quad \ldots \quad A^{(i)}_{1,n} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(1)}_{n,s,r}\mathbf{D}^{(s)}, \tag{5.16}$$

$$A^{(i)}_{2,1} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(2)}_{1,s,r}\mathbf{D}^{(s)}, \quad A^{(i)}_{2,2} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(2)}_{2,s,r}\mathbf{D}^{(s)}, \quad \ldots \quad A^{(i)}_{2,n} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(2)}_{n,s,r}\mathbf{D}^{(s)}, \tag{5.17}$$

$$\vdots$$

$$A^{(i)}_{n,1} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(n)}_{1,s,r}\mathbf{D}^{(s)}, \quad A^{(i)}_{n,2} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(n)}_{2,s,r}\mathbf{D}^{(s)}, \quad \ldots \quad A^{(i)}_{n,n} = \sum_{s=0}^{p}\boldsymbol{\alpha}^{(n)}_{n,s,r}\mathbf{D}^{(s)}, \tag{5.18}$$

and

$$\boldsymbol{\alpha}_{n,s,r}^{(n)} = \begin{bmatrix} \alpha_{n,s,r}^{(n)}(\eta_0,\zeta_j) & & & \\ & \alpha_{n,s,r}^{(n)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \alpha_{n,s,r}^{(n)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.19}$$

$$\boldsymbol{\beta}_{v,r}^{(k)} = \begin{bmatrix} \beta_{v,r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \beta_{v,r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \beta_{v,r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.20}$$

$$\boldsymbol{\gamma}_{v,r}^{(k)} = \begin{bmatrix} \gamma_{v,r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \gamma_{v,r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \gamma_{v,r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}. \tag{5.21}$$

Equation (5.15), is expressed as

$$\mathbf{B}_r \Omega_{r+1} = \mathbf{R}_r \tag{5.22}$$

where the coefficient matrix $\mathbf{B}_r$ is defined as

$$\left[\begin{array}{cccc|cccc|ccc|cccc}
B_{1,1}^{(0,0)} & B_{1,2}^{(0,0)} & \cdots & B_{1,n}^{(0,0)} & B_{1,1}^{(0,1)} & B_{1,2}^{(0,1)} & \cdots & B_{1,n}^{(0,1)} & \ddots & & & B_{1,1}^{(0,N_\eta)} & B_{1,2}^{(0,N_\eta)} & \cdots & B_{1,n}^{(0,N_\eta)} \\
B_{2,1}^{(0,0)} & B_{2,2}^{(0,0)} & \cdots & B_{2,n}^{(0,0)} & B_{2,1}^{(0,1)} & B_{2,2}^{(0,1)} & \cdots & B_{2,n}^{(0,1)} & & \ddots & & B_{2,1}^{(0,N_\eta)} & B_{2,2}^{(0,N_\eta)} & \cdots & B_{2,n}^{(0,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(0,0)} & B_{n,2}^{(0,0)} & \cdots & B_{n,n}^{(0,0)} & B_{n,1}^{(0,1)} & B_{n,2}^{(0,1)} & \cdots & B_{n,n}^{(0,1)} & & & \ddots & B_{n,1}^{(0,N_\eta)} & B_{n,2}^{(0,N_\eta)} & \cdots & B_{n,n}^{(0,N_\eta)} \\
\hline
B_{1,1}^{(1,0)} & B_{1,2}^{(1,0)} & \cdots & B_{1,n}^{(1,0)} & B_{1,1}^{(1,1)} & B_{1,2}^{(1,1)} & \cdots & B_{1,n}^{(1,1)} & \ddots & & & B_{1,1}^{(1,N_\eta)} & B_{1,2}^{(1,N_\eta)} & \cdots & B_{1,n}^{(1,N_\eta)} \\
B_{2,1}^{(1,0)} & B_{2,2}^{(1,0)} & \cdots & B_{2,n}^{(1,0)} & B_{2,1}^{(1,1)} & B_{2,2}^{(1,1)} & \cdots & B_{2,n}^{(1,1)} & & \ddots & & B_{2,1}^{(1,N_\eta)} & B_{2,2}^{(1,N_\eta)} & \cdots & B_{2,n}^{(1,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(1,0)} & B_{n,2}^{(1,0)} & \cdots & B_{n,n}^{(1,0)} & B_{n,1}^{(1,1)} & B_{n,2}^{(1,1)} & \cdots & B_{n,n}^{(1,1)} & & & \ddots & B_{n,1}^{(1,N_\eta)} & B_{n,2}^{(1,N_\eta)} & \cdots & B_{n,n}^{(1,N_\eta)} \\
\hline
\ddots & & & & \ddots & & & & \cdots & \cdots & \cdots & \ddots & & & \\
& \ddots & & & & \ddots & & & \cdots & \cdots & \cdots & & \ddots & & \\
& & \ddots & & & & \ddots & & \cdots & \cdots & \cdots & & & \ddots & \\
& & & \ddots & & & & \ddots & \cdots & \cdots & \cdots & & & & \ddots \\
\hline
B_{1,1}^{(N_\eta,0)} & B_{1,2}^{(N_\eta,0)} & \cdots & B_{1,n}^{(N_\eta,0)} & B_{1,1}^{(N_\eta,1)} & B_{1,2}^{(N_\eta,1)} & \cdots & B_{1,n}^{(N_\eta,1)} & \ddots & & & B_{1,1}^{(N_\eta,N_\eta)} & B_{1,2}^{(N_\eta,N_\eta)} & \cdots & B_{1,n}^{(N_\eta,N_\eta)} \\
B_{2,1}^{(N_\eta,0)} & B_{2,2}^{(N_\eta,0)} & \cdots & B_{2,n}^{(N_\eta,0)} & B_{2,1}^{(N_\eta,1)} & B_{2,2}^{(N_\eta,1)} & \cdots & B_{2,n}^{(N_\eta,1)} & & \ddots & & B_{2,1}^{(N_\eta,N_\eta)} & B_{2,2}^{(N_\eta,N_\eta)} & \cdots & B_{2,n}^{(N_\eta,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(N_\eta,0)} & B_{n,2}^{(N_\eta,0)} & \cdots & B_{n,n}^{(N_\eta,0)} & B_{n,1}^{(N_\eta,1)} & B_{n,2}^{(N_\eta,1)} & \cdots & B_{n,n}^{(N_\eta,1)} & & & \ddots & B_{n,1}^{(N_\eta,N_\eta)} & B_{n,2}^{(N_\eta,N_\eta)} & \cdots & B_{n,n}^{(N_\eta,N_\eta)}
\end{array}\right]$$

and the entries are defined as

$$\begin{aligned}
B_{k,k}^{(i,i)} &= A_{v,k}^{i} + \boldsymbol{\beta}_{v,r}^{(1,k)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_{v,k}^{(1,k)} d_{i,i}\mathbf{D}, \quad \text{for } v,k = 1,2,\ldots,n, \text{ when } i = j, \\
B_{k,k}^{(i,j)} &= \boldsymbol{\beta}_{v,r}^{(1,k)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_{v,k}^{(1,k)} d_{i,j}\mathbf{D}, \quad \text{for } v,k = 1,2,\ldots,n, \text{ when } i \neq j
\end{aligned} \tag{5.23}$$

The vectors Ω_{r+1} and $\mathbf{R}_r$ are defined as

$$\Omega_{r+1} = \left[\mathbf{F}_{1,r+1}^{(0)}\mathbf{F}_{2,r+1}^{(0)}\cdots\mathbf{F}_{n,r+1}^{(0)} \,\middle|\, \mathbf{F}_{1,r+1}^{(1)}\mathbf{F}_{2,r+1}^{(1)}\cdots\mathbf{F}_{n,r+1}^{(1)} \,\middle|\, \cdots\cdots\cdots\cdots \,\middle|\, \mathbf{F}_{1,r+1}^{(N_\zeta)}\mathbf{F}_{2,r+1}^{(N_\zeta)}\cdots\mathbf{F}_{n,r+1}^{(N_\zeta)}\right]^T \tag{5.24}$$

$$\mathbf{R}_r = \left[\mathbf{R}_1^{(0)}\ \mathbf{R}_2^{(0)}\ \mathbf{R}_3^{(0)}\cdots\mathbf{R}_n^{(0)} \,\middle|\, \mathbf{R}_1^{(1)}\ \mathbf{R}_2^{(1)}\ \mathbf{R}_3^{(1)}\cdots\mathbf{R}_n^{(1)} \,\middle|\, \cdots\cdots\cdots\cdots \,\middle|\, \mathbf{R}_1^{(N_\zeta)}\ \mathbf{R}_2^{(N_\zeta)}\ \cdots\ \mathbf{R}_n^{(N_\zeta)}\right]^T \tag{5.25}$$

Bivariate Spectral Local Linearisation Method

In this section we introduce the bivariate spectral local linearisation method (BSLLM) for approximating solutions of system of nonlinear partial differential equations. We develop

the bivariate spectral local linearization method for a general system of n nonlinear partial differential equations. Without loss of generality, consider equations of the form (5.1). The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$f_k(\eta,\zeta) \approx \sum_{i=0}^{N_\eta} \sum_{j=0}^{N_\zeta} f_k(\eta_i,\zeta_j)\mathscr{L}_i(\eta)\mathscr{L}_j(\zeta), \tag{5.26}$$

for $k = 1,2,\ldots,n$. The grid points are given by equation (1.9) and the function $\mathscr{L}_i(\eta)$ is the characteristic Lagrange cardinal polynomial given by equation (1.5). Applying the quasilinearisation method independently in each equation, we get

$$\begin{aligned}
&H_{1,r+1}\cdot\nabla_{f_1}\Gamma_1\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right] = H_{1,r}\cdot\nabla_{f_1}\Gamma_1\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right]-\Gamma_1\left[H_{1,r},H_{2,r},\ldots,H_{n,r}\right]\\
&H_{2,r+1}\cdot\nabla_{f_2}\Gamma_2\left[H_{1,r+1},H_{2,r},\ldots,H_{n,r}\right] = H_{2,r}\cdot\nabla_{f_2}\Gamma_2\left[H_{1,r+1},H_{2,r},\ldots,H_{n,r}\right]-\Gamma_2\left[H_{1,r+1},H_{2,r},\ldots,H_{n,r}\right]\\
&\vdots\\
&H_{n,r+1}\cdot\nabla_{f_n}\Gamma_n\left[H_{1,r+1},H_{2,r+1},\ldots,H_{n-1,r+1},H_{n,r}\right] = H_{n,r}\cdot\nabla_{f_n}\Gamma_n\left[H_{1,r+1},H_{2,r+1},\ldots,H_{n-1,r+1},H_{n,r}\right]\\
&-\Gamma_n\left[H_{1,r+1},H_{2,r+1},\ldots,H_{n-1,r+1},H_{n,r}\right].
\end{aligned} \tag{5.27}$$

Equation (5.27) form a system of n decoupled linear partial differential equations. They are solved iteratively for $f_1(\eta,\zeta), f_2(\eta,\zeta)),\ldots,f_n(\eta,\zeta))$. Equation (5.27) can further be expressed as:

$$\begin{aligned}
&\sum_{s=0}^{p}\alpha_{s,r}^{(1)}(\eta,\zeta)f_{1,r+1}^{(s)}+\beta_r^{(1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(0)}}{\partial\zeta}+\gamma_r^{(1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(1)}}{\partial\zeta} = R_1(\eta,\zeta),\\
&\sum_{s=0}^{p}\alpha_{s,r}^{(2)}(\eta,\zeta)f_{2,r+1}^{(s)}+\beta_r^{(2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(0)}}{\partial\zeta}+\gamma_r^{(2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(1)}}{\partial\zeta} = R_2(\eta,\zeta),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&\sum_{s=0}^{p}\alpha_{s,r}^{(n)}(\eta,\zeta)f_{n,r+1}^{(s)}+\beta_r^{(n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(0)}}{\partial\zeta}+\gamma_r^{(n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(1)}}{\partial\zeta} = R_n(\eta,\zeta),
\end{aligned} \tag{5.28}$$

where $\alpha_{s,r}^{(k)}(\eta,\zeta)$, $\beta_r^{(k)}(\eta,\zeta)$ and $\gamma_r^{(k)}(\eta,\zeta)$ are variable coefficients of $f_{k,r+1}^{(s)}$, $\frac{\partial f_{k,r+1}^{(0)}}{\partial\zeta}$, and $\frac{\partial f_{k,r+1}^{(1)}}{\partial\zeta}$, respectively, for $k = 1,2,\ldots,n$ and $s = 0,1,2,\ldots,p$. These coefficients correspond to the k^{th}

equation, for $k = 1, 2, \ldots, n$. Since constant p denotes the order of differentiation, then

$$\alpha_{s,r}^{(k)}(\eta,\zeta) = \frac{\partial \Gamma_k}{\partial f_{k,r}^{(s)}}, \qquad \beta_r^{(k)}(\eta,\zeta) = \frac{\partial \Gamma_k}{\partial \left(\frac{\partial f_{k,r}^{(0)}}{\partial \zeta}\right)}, \qquad \gamma_r^{(k)}(\eta,\zeta) = \frac{\partial \Gamma_k}{\partial \left(\frac{\partial f_{k,r}^{(1)}}{\partial \zeta}\right)}. \tag{5.29}$$

The kth right hand side is generally given by

$$R_k(\eta,\zeta) = \sum_{s=0}^{p} \alpha_{s,r}^{(k)}(\eta,\zeta) f_{k,r}^{(s)} + \beta_r^{(k)}(\eta,\zeta)\frac{\partial f_{k,r}^{(0)}}{\partial \zeta} + \gamma_r^{(k)}(\eta,\zeta)\frac{\partial f_{k,r}^{(1)}}{\partial \zeta)} - \Gamma_k. \tag{5.30}$$

Evaluating equation (5.28), at the Chebyshev-Gauss-Lobbatto grid points ζ_j ($j = 0, 1, \ldots, N_\zeta$) and η_i ($i = 0, 1, \ldots, N_\eta$), yields

$$\begin{aligned}
A_{1,1}\mathbf{F}_{1,i} + \boldsymbol{\beta}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{1,j} + \boldsymbol{\gamma}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{1,j} &= \mathbf{R}_{1,i}, \\
A_{2,2}\mathbf{F}_{2,i} + \boldsymbol{\beta}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{2,j} + \boldsymbol{\gamma}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{2,j} &= \mathbf{R}_{2,i}, \\
&\vdots \\
A_{n,n}\mathbf{F}_{n,i} + \boldsymbol{\beta}_r^{(n)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{n,j} + \boldsymbol{\gamma}_r^{(n)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{n,j} &= \mathbf{R}_{n,i},
\end{aligned} \tag{5.31}$$

where

$$A_{1,1} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(1)}\mathbf{D}^{(s)}, \quad A_{2,2} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(2)}\mathbf{D}^{(s)}, \quad \ldots, \quad A_{n,n} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(n)}\mathbf{D}^{(s)}. \tag{5.32}$$

The diagonal matrices of the corresponding variable coefficients at each kth equations are given by

$$\boldsymbol{\alpha}_{s,r}^{(k)} = \begin{bmatrix} \alpha_{s,r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \alpha_{s,r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \alpha_{s,r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.33}$$

$$\boldsymbol{\beta}_{r}^{(k)} = \begin{bmatrix} \beta_{r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \beta_{r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \beta_{r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.34}$$

$$\boldsymbol{\gamma}_{r}^{(k)} = \begin{bmatrix} \gamma_{r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \gamma_{r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \gamma_{r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}. \tag{5.35}$$

Imposing boundary conditions for $i = 0,1,\cdots,N_\zeta - 1$, equation (5.31), can be expressed as the following $N_\zeta(N_\eta+1)\times N_\zeta(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0}^{(k)} & B_{0,1}^{(k)} & \cdots & B_{0,N_\zeta-1}^{(k)} \\ B_{1,0}^{(k)} & B_{1,1}^{(k)} & \cdots & B_{1,N_\zeta-1}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(k)} & B_{N_\zeta-1,1}^{(k)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{k,0} \\ \mathbf{F}_{k,1} \\ \vdots \\ \mathbf{F}_{k,N_\zeta-1} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{k,0} \\ \mathscr{R}_{k,1} \\ \vdots \\ \mathscr{R}_{k,N_\zeta-1} \end{bmatrix}, \tag{5.36}$$

where

$$\begin{aligned} B_{(i,i)}^{(k)} &= \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(k)}\mathbf{D}^{(s)} + \boldsymbol{\beta}_{r}^{(k)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,i}\mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \text{ when } i = j, \\ B_{(i,j)}^{(k)} &= \boldsymbol{\beta}_{r}^{(k)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,j}\mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \text{ when } i \neq j. \end{aligned} \tag{5.37}$$

The vector $\mathscr{R}_{k,i}$ is defined as

$$\mathscr{R}_{k,i} = \mathbf{R}_{k,i} - \left(\boldsymbol{\beta}_{r}^{(k)} d_{i,N_\zeta}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,N_\zeta}\mathbf{D}\right)\mathbf{F}_{k,N_\zeta}$$

for $i = 0, 1, \cdots, N_\zeta - 1$ and $k = 1, 2, \ldots, n$. The vector $\mathbf{F}_{k,N_\zeta}$ corresponds to the initial boundary condition which is always prescribed.

Bivariate Spectral Relaxation Method

The bivariate spectral relaxation method (BSRM) for approximating solutions of system of nonlinear partial differential equations is developed for a general system of n nonlinear partial differential equations. Similarly, without loss of generality, equations of the form (5.1) are considered. The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form (5.26). The algorithm for the method is summarized as follows:

1. The iteration scheme is developed by assuming that in the first equation, only the linear terms are evaluated at the current iteration which is denoted by $r+1$ and all the other terms irrespective of their linearity are evaluated at the previous iteration denoted by r. These terms and all derivatives in terms of the first equation are assumed to be known from previous iteration.

2. For the second equation, the linear terms are evaluated at the current iteration and the other terms are evaluated at the previous iteration. This procedure is repeated until the last equation.

We should remark that the BSRM method uses a similar approach to the BSLLM method except that the BSRM method does not use the quasilinearisation approach but rather the Gauss-Seidel approach to decouple the equations and hence resulting in different variable coefficients from the BSLLM. The BSRM for systems of three equations or more have been applied to various problems by many researchers including [132, 139]. Rearranging terms in equation (5.27), we

obtain

$$\begin{aligned}
&\sum_{s=0}^{p} \alpha_{s,r}^{(1)}(\eta,\zeta) f_{1,r+1}^{(s)} + \beta_r^{(1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(0)}}{\partial \zeta} + \gamma_r^{(1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(1)}}{\partial \zeta} = R_1(\eta,\zeta),\\
&\sum_{s=0}^{p} \alpha_{s,r}^{(2)}(\eta,\zeta) f_{2,r+1}^{(s)} + \beta_r^{(2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(0)}}{\partial \zeta} + \gamma_r^{(2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(1)}}{\partial \zeta} = R_2(\eta,\zeta),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&\sum_{s=0}^{p} \alpha_{s,r}^{(n)}(\eta,\zeta) f_{n,r+1}^{(s)} + \beta_r^{(n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(0)}}{\partial \zeta} + \gamma_r^{(n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(1)}}{\partial \zeta} = R_n(\eta,\zeta),
\end{aligned} \tag{5.38}$$

where $\alpha_{s,r}^{(k)}(\eta,\zeta)$, $\beta_r^{(k)}(\eta,\zeta)$ and $\gamma_r^{(k)}(\eta,\zeta)$ are variable coefficients of $f_{k,r+1}^{(s)}$, $\frac{\partial f_{k,r+1}^{(0)}}{\partial \zeta}$, and $\frac{\partial f_{k,r+1}^{(1)}}{\partial \zeta}$, respectively, for $k = 1,2,\ldots,n$ and $s = 0,1,2,\ldots,p$. These coefficients correspond to the k^{th} equation, for $k = 1,2,\ldots,n$. The right hand side of the kth equation depends on the number of linear and nonlinear terms of the kth equation and other equations other than the kth equation. Evaluating equation (5.38), at the Chebyshev-Gauss-Lobbatto grid points ζ_j ($j = 0,1,\ldots,N_\zeta$) and η_i ($i = 0,1,\ldots,N_\eta$), yields

$$\begin{aligned}
&A_{1,1}\mathbf{F}_{1,i} + \boldsymbol{\beta}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{1,j} + \boldsymbol{\gamma}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{1,j} = \mathbf{R}_{1,i},\\
&A_{2,2}\mathbf{F}_{2,i} + \boldsymbol{\beta}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{2,j} + \boldsymbol{\gamma}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{2,j} = \mathbf{R}_{2,i},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&A_{n,n}\mathbf{F}_{n,i} + \boldsymbol{\beta}_r^{(n)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{n,j} + \boldsymbol{\gamma}_r^{(n)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{n,j} = \mathbf{R}_{n,i},
\end{aligned} \tag{5.39}$$

where

$$A_{1,1} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(1)}\mathbf{D}^{(s)}, \quad A_{2,2} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(2)}\mathbf{D}^{(s)}, \quad \ldots, \quad A_{n,n} = \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(n)}\mathbf{D}^{(s)}. \tag{5.40}$$

The variable coefficients give rise to diagonal matrices. The corresponding diagonal matrices of each kth equation are given by

$$\boldsymbol{\alpha}_{s,r}^{(k)} = \begin{bmatrix} \alpha_{s,r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \alpha_{s,r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \alpha_{s,r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.41}$$

$$\boldsymbol{\beta}_{r}^{(k)} = \begin{bmatrix} \beta_{r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \beta_{r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \beta_{r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}, \tag{5.42}$$

$$\boldsymbol{\gamma}_{r}^{(k)} = \begin{bmatrix} \gamma_{r}^{(k)}(\eta_0,\zeta_j) & & & \\ & \gamma_{r}^{(k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \gamma_{r}^{(k)}(\eta_{N_\eta},\zeta_j) \end{bmatrix}. \tag{5.43}$$

Imposing boundary conditions for $i = 0,1,\cdots,N_\zeta - 1$, equation (5.39), can be expressed as the following $N_\zeta(N_\eta+1)\times N_\zeta(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0}^{(k)} & B_{0,1}^{(k)} & \cdots & B_{0,N_\zeta-1}^{(k)} \\ B_{1,0}^{(k)} & B_{1,1}^{(k)} & \cdots & B_{1,N_\zeta-1}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(k)} & B_{N_\zeta-1,1}^{(k)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{k,0} \\ \mathbf{F}_{k,1} \\ \vdots \\ \mathbf{F}_{k,N_\zeta-1} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{k,0} \\ \mathscr{R}_{k,1} \\ \vdots \\ \mathscr{R}_{k,N_\zeta-1} \end{bmatrix}, \tag{5.44}$$

where

$$\begin{aligned} B_{(i,i)}^{(k)} &= \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(k)}\mathbf{D}^{(s)} + \boldsymbol{\beta}_{r}^{(k)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,i}\mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \text{ when } i = j, \\ B_{(i,j)}^{(k)} &= \boldsymbol{\beta}_{r}^{(k)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,j}\mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \text{ when } i \neq j. \end{aligned} \tag{5.45}$$

The vector $\mathscr{R}_{k,i}$ is defined as

$$\mathscr{R}_{k,i} = \mathbf{R}_{k,i} - \left(\boldsymbol{\beta}_{r}^{(k)} d_{i,N_\zeta}\mathbf{I} + \boldsymbol{\gamma}_{r}^{(k)} d_{i,N_\zeta}\mathbf{D}\right)\mathbf{F}_{k,N_\zeta}$$

for $i = 0,1,\cdots,N_\zeta - 1$ and $k = 1,2,\ldots,n$. Similarly, the vector $\mathbf{F}_{k,N_\zeta}$ corresponds to the initial boundary condition which is always prescribed.

5.3 Numerical Experiments

In this subsection, we present a system of three nonlinear coupled partial differential equations that model an unsteady three-dimensional MHD-boundary-layer flow due to an impulsive motion of a stretching surface. The equations are given by:

$$\begin{aligned}
&\frac{\partial^3 f}{\partial \eta^3} + \frac{1}{2}\eta(1-\zeta)\frac{\partial^2 f}{\partial \eta^2} + \zeta(f+s)\frac{\partial^2 f}{\partial \eta^2} - \zeta\left(\frac{\partial f}{\partial \eta}\right)^2 - M\zeta\frac{\partial f}{\partial \eta} = \zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial f}{\partial \eta}\right),\\
&\frac{\partial^3 s}{\partial \eta^3} + \frac{1}{2}\eta(1-\zeta)\frac{\partial^2 s}{\partial \eta^2} + \zeta(f+s)\frac{\partial^2 s}{\partial \eta^2} - \zeta\left(\frac{\partial s}{\partial \eta}\right)^2 - M\zeta\frac{\partial s}{\partial \eta} = \zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial s}{\partial \eta}\right),\\
&\frac{1}{Pr}\frac{\partial^2 g}{\partial \eta^2} + \frac{1}{2}\eta(1-\zeta)\frac{\partial g}{\partial \eta} + \zeta(f+s)\frac{\partial g}{\partial \eta} = \zeta(1-\zeta)\frac{\partial g}{\partial \zeta}.
\end{aligned} \tag{5.46}$$

The appropriate boundary conditions are given by

$$\begin{aligned}
f(\zeta,0) = 0, \quad f'(\zeta,0) = 1, \quad s(\zeta,0) = 0, \quad g(\zeta,0) = 1, \quad s'(\zeta,0) = c,\\
f'(\zeta,\infty) = s'(\zeta,\infty) = f(\zeta,\infty) = 0.
\end{aligned} \tag{5.47}$$

In this model, c is the ratio f the surface velocity gradients along the y and x directions, Pr is the Prandtl number and M is the magnetic number. This model was formulated and modeled by Takhar et. al. [140] using finite differences. It has been solved by other numerical methods by different researchers. We solve the same model using the bivariate spectral quasilinearisation method (BSQLM), bivariate spectral relaxation method (BSRM) and bivariate spectral local linearisation method (BSLLM) in the subsequent subsections. The exact solutions for the equations when $\zeta = 0$ are given by:

$$\begin{aligned}
f(\eta) &= \eta\,\mathrm{erfc}\left(\frac{\eta}{2}\right) + \frac{1}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right],\\
s(\eta) &= c\eta\,\mathrm{erfc}\left(\frac{\eta}{2}\right) + \frac{1}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right],\\
g(\eta) &= \mathrm{erfc}\left(\frac{\sqrt{Pr}\eta}{2}\right).
\end{aligned} \tag{5.48}$$

We express the governing equation (5.46) in the general form (5.1). Therefore, we have

$$\begin{aligned}
&\frac{\partial^3 f_1}{\partial \eta^3}+\frac{1}{2}\eta(1-\zeta)\frac{\partial^2 f_1}{\partial \eta^2}+\zeta(f_1+f_2)\frac{\partial^2 f_1}{\partial \eta^2}-\zeta\left(\frac{\partial f_1}{\partial \eta}\right)^2-M\zeta\frac{\partial f_1}{\partial \eta}-\zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial f_1}{\partial \eta}\right)=0,\\
&\frac{\partial^3 f_2}{\partial \eta^3}+\frac{1}{2}\eta(1-\zeta)\frac{\partial^2 f_2}{\partial \eta^2}+\zeta(f_1+f_2)\frac{\partial^2 f_2}{\partial \eta^2}-\zeta\left(\frac{\partial f_2}{\partial \eta}\right)^2-M\zeta\frac{\partial f_2}{\partial \eta}-\zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial f_2}{\partial \eta}\right)=0,\\
&\frac{1}{Pr}\frac{\partial^2 f_3}{\partial \eta^2}+\frac{1}{2}\eta(1-\zeta)\frac{\partial f_3}{\partial \eta}+\zeta(f_1+f_2)\frac{\partial f_3}{\partial \eta}-\zeta(1-\zeta)\frac{\partial f_3}{\partial \zeta}=0.
\end{aligned} \tag{5.49}$$

subject to

$$\begin{aligned}
f_1(\zeta,0)=0,\quad f_1'(\zeta,0)=1,\quad f_2(\zeta,0)=0,\quad f_3(\zeta,0)=1,\quad f_2'(\zeta,0)=c,\\
f_1'(\zeta,\infty)=f_2'(\zeta,\infty)=f_3(\zeta,\infty)=0.
\end{aligned} \tag{5.50}$$

We define

$$\begin{aligned}
\Gamma_1&=\frac{\partial^3 f_1}{\partial \eta^3}+\frac{1}{2}\eta(1-\zeta)\frac{\partial^2 f_1}{\partial \eta^2}+\zeta(f_1+f_2)\frac{\partial^2 f_1}{\partial \eta^2}-\zeta\left(\frac{\partial f_1}{\partial \eta}\right)^2-M\zeta\frac{\partial f_1}{\partial \eta}-\zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial f_1}{\partial \eta}\right),\\
\Gamma_2&=\frac{\partial^3 f_2}{\partial \eta^3}+\frac{1}{2}\eta(1-\zeta)\frac{\partial^2 f_2}{\partial \eta^2}+\zeta(f_1+f_2)\frac{\partial^2 f_2}{\partial \eta^2}-\zeta\left(\frac{\partial f_2}{\partial \eta}\right)^2-M\zeta\frac{\partial f_2}{\partial \eta}-\zeta(1-\zeta)\frac{\partial}{\partial \zeta}\left(\frac{\partial f_2}{\partial \eta}\right),\\
\Gamma_3&=\frac{1}{Pr}\frac{\partial^2 f_3}{\partial \eta^2}+\frac{1}{2}\eta(1-\zeta)\frac{\partial f_3}{\partial \eta}+\zeta(f_1+f_2)\frac{\partial f_3}{\partial \eta}-\zeta(1-\zeta)\frac{\partial f_3}{\partial \zeta}.
\end{aligned} \tag{5.51}$$

Bivariate spectral quasilinearisation method (BSQLM)

In this subsection, we demonstrate the application of the BSQLM method on the numerical experiment. The highest order of differentiation is $p=3$ and the number of equations is $n=3$. Applying the BSQLM method to equation (5.49), we get

$$\begin{aligned}
\sum_{v=1}^{3}\left[A_{1,v}^{(i)}\mathbf{F}_{v,i}+\boldsymbol{\beta}_{v,r}^{(1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}+\boldsymbol{\gamma}_{v,r}^{(1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}\right]&=\mathbf{R}_{1,i},\\
\sum_{v=1}^{3}\left[A_{2,v}^{(i)}\mathbf{F}_{v,i}+\boldsymbol{\beta}_{v,r}^{(2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}+\boldsymbol{\gamma}_{v,r}^{(2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}\right]&=\mathbf{R}_{2,i},\\
\sum_{v=1}^{3}\left[A_{3,v}^{(i)}\mathbf{F}_{v,i}+\boldsymbol{\beta}_{v,r}^{(3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}+\boldsymbol{\gamma}_{v,r}^{(3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}\right]&=\mathbf{R}_{3,i},
\end{aligned} \tag{5.52}$$

where

$$A_{1,1}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{1,s,r}^{(1)} \mathbf{D}^{(s)}, \quad A_{1,2}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{2,s,r}^{(1)} \mathbf{D}^{(s)}, \quad A_{1,3}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{3,s,r}^{(1)} \mathbf{D}^{(s)}, \tag{5.53}$$

$$A_{2,1}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{1,s,r}^{(2)} \mathbf{D}^{(s)}, \quad A_{2,2}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{2,s,r}^{(2)} \mathbf{D}^{(s)}, \quad A_{2,3}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{3,s,r}^{(2)} \mathbf{D}^{(s)}, \tag{5.54}$$

$$A_{3,1}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{1,s,r}^{(3)} \mathbf{D}^{(s)}, \quad A_{3,2}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{2,s,r}^{(3)} \mathbf{D}^{(s)}, \quad A_{3,3}^{(i)} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{3,s,r}^{(3)} \mathbf{D}^{(s)}. \tag{5.55}$$

The coefficients for $s = 0, 1, 2, 3$ are given by

$$\alpha_{1,s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial f_{1,r}^{(s)}}, \quad \alpha_{2,s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial f_{2,r}^{(s)}}, \quad \alpha_{3,s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial f_{3,r}^{(s)}}, \quad \alpha_{1,s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial f_{1,r}^{(s)}}, \quad \alpha_{2,s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial f_{2,r}^{(s)}}, \quad \alpha_{3,s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial f_{3,r}^{(s)}},$$

$$\alpha_{1,s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial f_{1,r}^{(s)}}, \quad \alpha_{2,s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial f_{2,r}^{(s)}}, \quad \alpha_{3,s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial f_{3,r}^{(s)}}, \quad \beta_{s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial \left(\frac{\partial f_{s,r}}{\partial \zeta}\right)}, \quad \beta_{s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial \left(\frac{\partial f_{s,r}}{\partial \zeta}\right)},$$

$$\beta_{s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial \left(\frac{\partial f_{s,r}}{\partial \zeta}\right)}, \quad \gamma_{s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial \left(\frac{\partial}{\partial \zeta}\left(\frac{\partial f_{s,r}}{\partial \eta}\right)\right)}, \quad \gamma_{s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial \left(\frac{\partial}{\partial \zeta}\left(\frac{\partial f_{s,r}}{\partial \eta}\right)\right)}, \quad \gamma_{s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial \left(\frac{\partial}{\partial \zeta}\left(\frac{\partial f_{s,r}}{\partial \eta}\right)\right)}.$$

These information can be used to construct the elements of the coefficient matrix $\mathbf{B}_r$, which are given by equation (5.23).

Bivariate spectral local linearisation method (BSLLM)

Similarly, in this subsection, we demonstrate the application of the BSLLM method on the numerical experiment. The highest order of differentiation is $p = 3$ and $n = 3$. Applying the BSLLM method to equations (5.49), we get

$$\begin{aligned}
A_{1,1}\mathbf{F}_{1,i} + \boldsymbol{\beta}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{1,j} + \boldsymbol{\gamma}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{1,j} &= \mathbf{R}_{1,i}, \\
A_{2,2}\mathbf{F}_{2,i} + \boldsymbol{\beta}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{2,j} + \boldsymbol{\gamma}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{2,j} &= \mathbf{R}_{2,i}, \\
A_{3,3}\mathbf{F}_{3,i} + \boldsymbol{\beta}_r^{(3)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{3,j} + \boldsymbol{\gamma}_r^{(3)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{3,j} &= \mathbf{R}_{3,i},
\end{aligned} \tag{5.56}$$

where

$$A_{1,1} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(1)} \mathbf{D}^{(s)}, \quad A_{2,2} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(2)} \mathbf{D}^{(s)}, \quad A_{3,3} = \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(3)} \mathbf{D}^{(s)}. \tag{5.57}$$

The coefficients for $s = 0, 1, 2, 3$ are given by:

$$\alpha_{s,r}^{(1)} = \frac{\partial \Gamma_1}{\partial f_{1,r}^{(s)}}, \quad \alpha_{s,r}^{(2)} = \frac{\partial \Gamma_2}{\partial f_{2,r}^{(s)}}, \quad \alpha_{s,r}^{(3)} = \frac{\partial \Gamma_3}{\partial f_{3,r}^{(s)}}, \quad \beta_r^{(1)} = \frac{\partial \Gamma_1}{\partial \left(\frac{\partial f_{1,r}^{(0)}}{\partial \zeta} \right)}, \quad \beta_r^{(2)} = \frac{\partial \Gamma_2}{\partial \left(\frac{\partial f_{2,r}^{(0)}}{\partial \zeta} \right)},$$

$$\beta_r^{(3)} = \frac{\partial \Gamma_3}{\partial \left(\frac{\partial f_{3,r}^{(0)}}{\partial \zeta} \right)}, \quad \gamma_r^{(1)} = \frac{\partial \Gamma_1}{\partial \left(\frac{\partial f_{1,r}^{(1)}}{\partial \zeta} \right)}, \quad \gamma_r^{(2)} = \frac{\partial \Gamma_2}{\partial \left(\frac{\partial f_{2,r}^{(1)}}{\partial \zeta} \right)}, \quad \gamma_r^{(3)} = \frac{\partial \Gamma_3}{\partial \left(\frac{\partial f_{3,r}^{(1)}}{\partial \zeta} \right)}.$$

The right hand side for $k = 1, 2, 3$ is given by

$$R_{k,i} = \sum_{s=0}^{3} \alpha_{s,r}^{(k)}(\eta, \zeta) f_{k,r}^{(s)} + \beta_r^{(k)}(\eta, \zeta) \frac{\partial f_{k,r}^{(0)}}{\partial \zeta} + \gamma_r^{(k)}(\eta, \zeta) \frac{\partial f_{k,r}^{(1)}}{\partial \zeta} - \Gamma_k. \tag{5.58}$$

Thus the equations can be expressed as a matrix system as equation (5.36) and the entries of the matrix are given by equations (5.37). The resulting matrix is independently solved for the desired functions.

Bivariate spectral relaxation method (BSRM)

In this subsection, we demonstrate the application of the BSRM method of the numerical experiment. The highest order of differentiation is $p = 3$ in the first two equations and $p = 2$ in the third order equation. The constant $n = 3$. Applying the BSRM method to equations (5.49), we get

$$\begin{aligned}
A_{1,1}\mathbf{F}_{1,i} + \boldsymbol{\beta}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{1,j} + \boldsymbol{\gamma}_r^{(1)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{1,j} &= \mathbf{R}_{1,i}, \\
A_{2,2}\mathbf{F}_{2,i} + \boldsymbol{\beta}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{2,j} + \boldsymbol{\gamma}_r^{(2)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{2,j} &= \mathbf{R}_{2,i}, \\
A_{3,3}\mathbf{F}_{3,i} + \boldsymbol{\beta}_r^{(3)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{3,j} + \boldsymbol{\gamma}_r^{(3)} \sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{D}\mathbf{F}_{3,j} &= \mathbf{R}_{3,i},
\end{aligned} \tag{5.59}$$

where

$$A_{1,1}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(1)}\mathbf{D}^{(s)},\quad A_{2,2}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(2)}\mathbf{D}^{(s)},\quad A_{3,3}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(3)}\mathbf{D}^{(s)}. \tag{5.60}$$

The coefficients for $s=0,1,2,3$ are given by:

$$\begin{aligned}
&\alpha_{0,r}^{(1)}=0,\quad \alpha_{1,r}^{(1)}=-M\zeta,\quad \alpha_{2,r}^{(1)}=\frac{1}{2}\eta(1-\zeta)+\zeta(f_{1,r}+f_{2,r}),\\
&\alpha_{3,r}^{(1)}=1,\quad \beta_{r}^{(1)}=0,\quad \gamma_{r}^{(1)}=-\zeta(1-\zeta),\\
&\alpha_{0,r}^{(2)}=0,\quad \alpha_{1,r}^{(2)}=-M\zeta,\quad \alpha_{2,r}^{(2)}=\frac{1}{2}\eta(1-\zeta)+\zeta(f_{1,r}+f_{2,r}),\\
&\alpha_{3,r}^{(2)}=1,\quad \beta_{r}^{(2)}=0,\quad \gamma_{r}^{(2)}=-\zeta(1-\zeta),\\
&\alpha_{0,r}^{(3)}=0,\quad \alpha_{1,r}^{(3)}=\frac{1}{2}\eta(1-\zeta)+\zeta(f_{1,r}+f_{2,r}),\quad \alpha_{2,r}^{(3)}=\frac{1}{Pr},\\
&\alpha_{3,r}^{(3)}=0,\quad \beta_{r}^{(3)}=-\zeta(1-\zeta),\quad \gamma_{r}^{(3)}=0.
\end{aligned} \tag{5.61}$$

The right hand side for $k=1,2,3$ is given by

$$\begin{aligned}
R_{1,r}&=\zeta\left(\frac{\partial f_1}{\partial\eta}\right)^2,\\
R_{2,r}&=\zeta\left(\frac{\partial f_2}{\partial\eta}\right)^2,\\
R_{3,r}&=0.
\end{aligned} \tag{5.62}$$

Thus the equations can be expressed as a matrix system as equation (5.44) and the entries of the matrix are given by equations (5.45). The resulting matrix is independently solved for the desired functions.

5.4 Results and Discussion

In this section we present the numerical solutions of the unsteady three-dimensional MHD-boundary-layer flow due to an impulsive motion of stretching surface obtained using the bivariate spectral quasilinearisation method (BSQLM), bivariate spectral relaxation method (BSRM) and bivariate spectral local linearisation method (BSLLM) algorithms. The η domain was truncated to $\eta_\infty=10$ for all computations. Accurate results for all the quantities of physical interest were

obtained using this value of η. Graphs and tables displaying physical quantities of interest are presented to validate the accuracy, general performance of the three methods. All the results in this section were obtained using MATLAB 2013, and the constants $c = 0.5, Pr = 0.7$ and $M = 1$. We consider two cases, we have case I, where $N_\eta = 20$ and $N_\zeta = 10$, $N_\zeta = 20$. We also have case II, where $N_\eta = 80$ and $N_\zeta = 30$, $N_\zeta = 40$.

Case I: $N_\eta = 20$ and $N_\zeta = 10$, $N_\zeta = 20$

Figs. 5.1 - 5.15 were obtained by using the bivariate spectral methods, that is, using the BSQLM, BSRM, BSLLM. Figs. 5.1 and 5.2 show the second derivatives of $f(\eta,\zeta)$, $s(\eta,\zeta)$ respectively. These graphs were obtained using the three methods, BSQLM, BSRM, BSLLM. These values from the three different methods agree for all value of η. A similar trend is observed for all the remaining graphs (Figs. 5.3-5.6). This in turn implies that the three methods give accurate results.

Figs. 5.7 - 5.15 show the values of $f(\eta,\zeta)$, $s(\eta,\zeta)$ and $g(\eta,\zeta)$ in the entire domain, that is, for all ξ and η. We note that the methods give exactly the same values of $f(\eta,\zeta)$, $s(\eta,\zeta)$ and $g(\eta,\zeta)$. These results prove the accuracy of these methods. However, the main question is, which method gives accurate solutions with minimal computational time? The results in the tables answers this question.

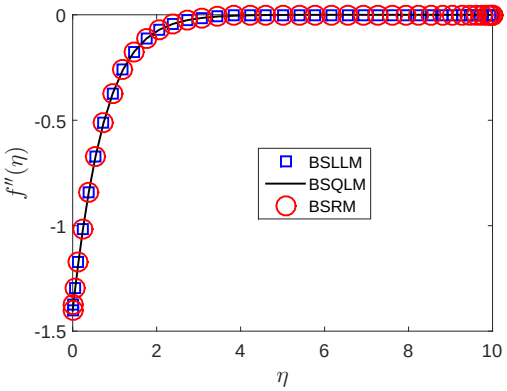

Fig. 5.1 Graph of $f''(\eta,\zeta)$

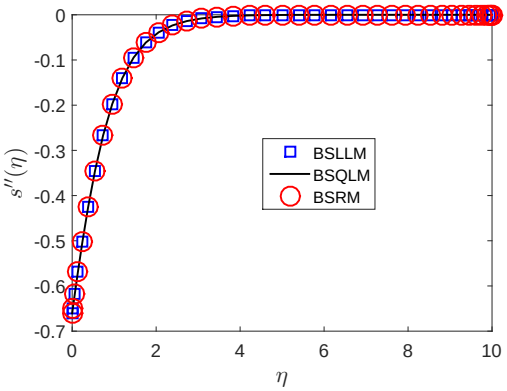

Fig. 5.2 Graph of $s''(\eta,\zeta)$

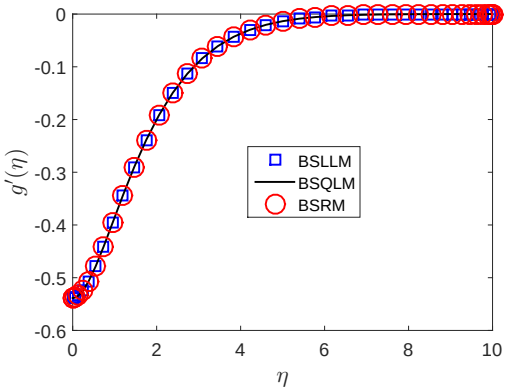

Fig. 5.3 Graph of $g'(\eta,\zeta)$

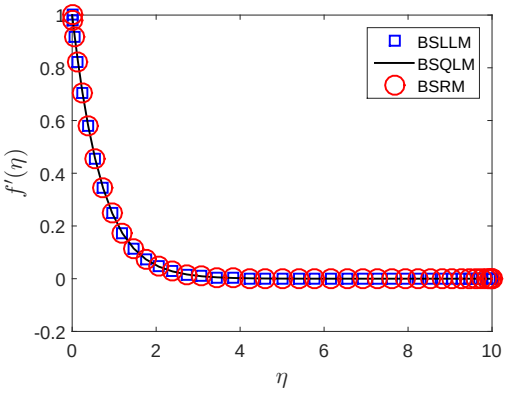

Fig. 5.4 Graph of $f'(\eta,\zeta)$

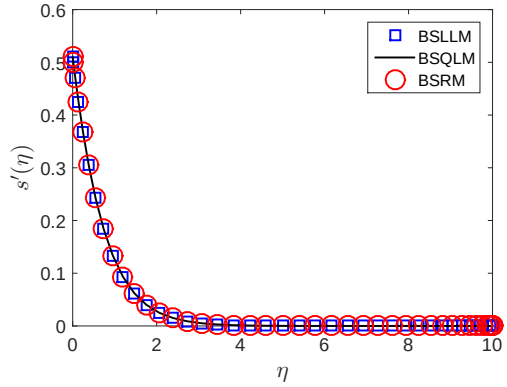

Fig. 5.5 Graph of $s'(\eta,\zeta)$

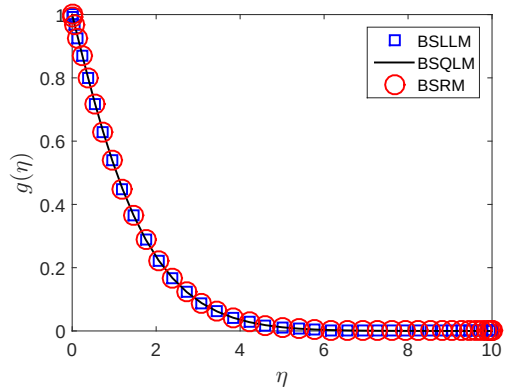

Fig. 5.6 Graph of $g(\eta,\zeta)$

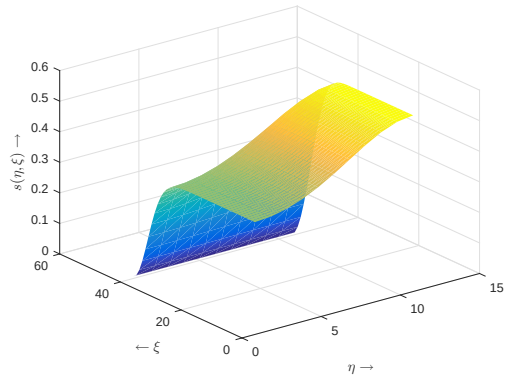

Fig. 5.7 Graph of $f(\eta,\zeta)$ using BSQLM

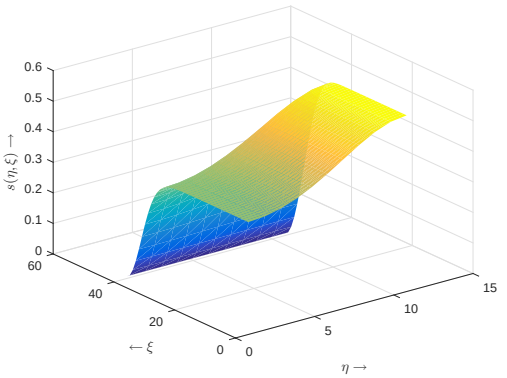

Fig. 5.8 Graph of $f(\eta,\zeta)$ using BSLLM

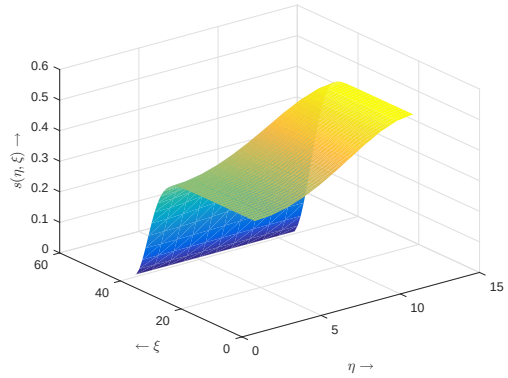

Fig. 5.9 Graph of $f(\eta,\zeta)$ using BSRM

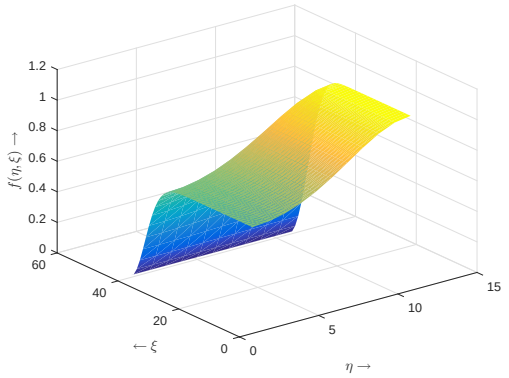

Fig. 5.10 Graph of $s(\eta,\zeta)$ using BSQLM

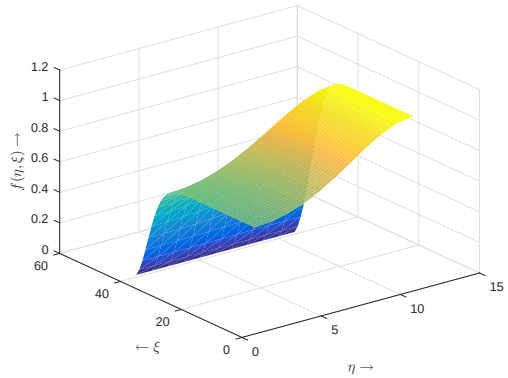

Fig. 5.11 Graph of $s(\eta,\zeta)$ using BSLLM

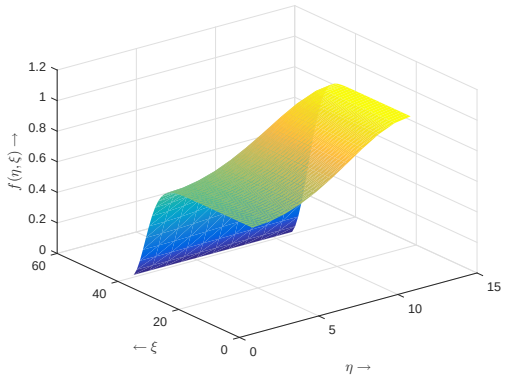

Fig. 5.12 Graph of $s(\eta,\zeta)$ using BSRM

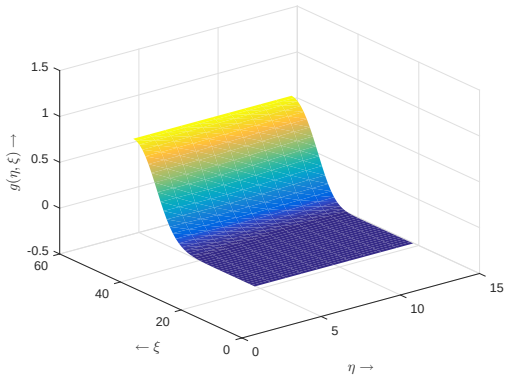

Fig. 5.13 Graph of $g(\eta,\zeta)$ using BSQLM

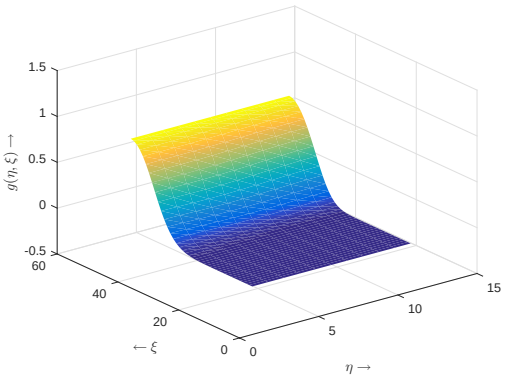

Fig. 5.14 Graph of $g(\eta,\zeta)$ using BSLLM

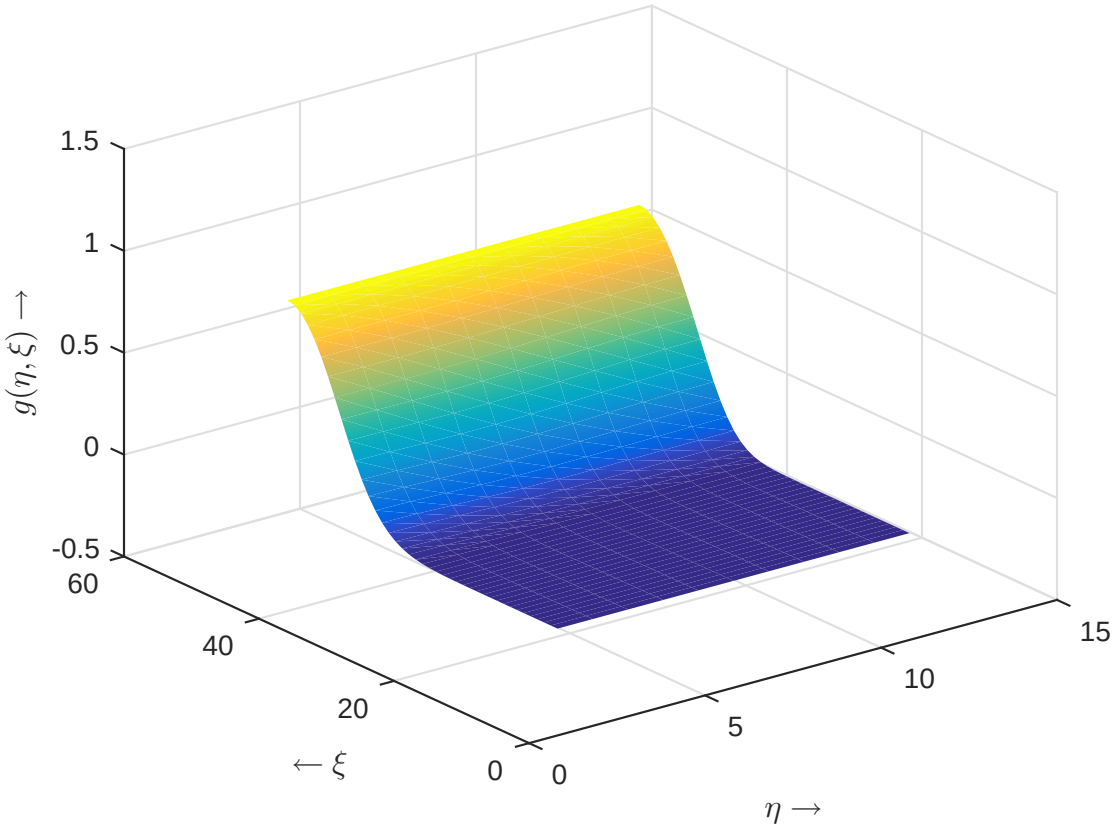

Fig. 5.15 Graph of $g(\eta,\zeta)$ using BSRM

In Table 5.1, the skin friction and Nusselt number are displayed. These results were generated using a fixed $N_\eta = 20$ and $N_\zeta = 10$, $N_\zeta = 20$. Increasing N_η did not improve the accuracy of the results but increased the computational time. Thus accurate results are obtained using few grid points and hence decreasing computational time. The BSLLM converged to the solution faster, followed by the BSRM and lastly, the BSQLM as evidenced by the computational time. We can conclude that the BSLLM method converges faster and hence achieves accuracy faster than the BSQLM and BSRM methods.

Table 5.1 The skin friction and Nusselt number when $c = 0.5, Pr = 0.7$ and $M = 1$.

	$N_\zeta = 10$			$N_\zeta = 20$		
ζ	BSLLM	BSRM	BSQLM	BSLLM	BSRM	BSQLM
	$f''(0,\zeta)$					
0.2	0.779746	0.779746	0.779746	0.779746	0.779746	0.779746
0.4	0.976713	0.976713	0.976713	0.976713	0.976713	0.976713
0.6	1.157305	1.157305	1.157305	1.157305	1.157305	1.157305
0.8	1.323432	1.323432	1.323432	1.323432	1.323432	1.323432
	$s''(0,\zeta)$					
0.2	0.371735	0.371735	0.371735	0.371735	0.371735	0.371735
0.4	0.456101	0.456101	0.456101	0.456101	0.456101	0.456101
0.6	0.53540	0.53540	0.53540	0.53540	0.53540	0.53540
0.8	0.609873	0.609873	0.609873	0.609873	0.609873	0.609873
	$g'(0,\zeta)$					
0.2	0.493606	0.493606	0.493606	0.493606	0.493606	0.493606
0.4	0.512696	0.512696	0.512696	0.512696	0.512696	0.512696
0.6	0.528264	0.528264	0.528264	0.528264	0.528264	0.528264
0.8	0.537671	0.537671	0.537671	0.537671	0.537671	0.537671
CPU time	0.131244	0.141244	0.940761	0.54191	0.673688	1.363688

Case II: $N_\eta = 80$ **and** $N_\zeta = 30$**,** $N_\zeta = 40$

In this section, we consider the case when $N_\eta = 80$ and $N_\zeta = 30$, $N_\zeta = 40$. We want to investigate about which method gives accurate results with minimal computational time. Table 5.2 shows the skin friction and Nusselt number. From the table, we can conclude that increasing N_η did not improve the accuracy of the results but increased the computational time. This implies that accurate results can be obtained using few grid points and hence decreasing computational time. The BSLLM converged to the solution faster, followed by the BSRM and lastly, the BSQLM as evidenced by the computational time. We can conclude that the BSLLM method converges faster and hence achieves accuracy faster than the BSQLM and BSRM methods. These results

are in line with the analysis we did in the previous Chapters of this thesis. It is important to note that the BSQLM method is second order accurate meanwhile the BSRM and BSLLM methods are first order accurate methods. The BSQLM method solves a coupled system of equations simultaneously meanwhile the BSRM and BSLLM methods decouple them first and solve them iteratively. This is one of the reasons why the BSQLM takes more computational time compared to the BSLLM and BSRM.

Table 5.2 The skin friction and Nusselt number when $c = 0.5, Pr = 0.7$ and $M = 1$.

	$N_\zeta = 30$			$N_\zeta = 40$		
ζ	BSLLM	BSRM	BSQLM	BSLLM	BSRM	BSQLM
	$f''(0,\zeta)$					
0.2	0.779746	0.779746	0.779746	0.779746	0.779746	0.779746
0.4	0.976713	0.976713	0.976713	0.976713	0.976713	0.976713
0.6	1.157305	1.157305	1.157305	1.157305	1.157305	1.157305
0.8	1.323432	1.323432	1.323432	1.323432	1.323432	1.323432
	$s''(0,\zeta)$					
0.2	0.371735	0.371735	0.371735	0.371735	0.371735	0.371735
0.4	0.456101	0.456101	0.456101	0.456101	0.456101	0.456101
0.6	0.53540	0.53540	0.53540	0.53540	0.53540	0.53540
0.8	0.609873	0.609873	0.609873	0.609873	0.609873	0.609873
	$g'(0,\zeta)$					
0.2	0.493606	0.493606	0.493606	0.493606	0.493606	0.493606
0.4	0.512696	0.512696	0.512696	0.512696	0.512696	0.512696
0.6	0.528264	0.528264	0.528264	0.528264	0.528264	0.528264
0.8	0.537671	0.537671	0.537671	0.537671	0.537671	0.537671
CPU time	1.533409	1.644579	1.943781	2.093979	2.127654	3.3492391

It is evident from both tables that increasing the number of grid points does not improve the accuracy of the methods but rather it increases the computational time.

5.5 Conclusion

In this chapter, we presented a generalized approach for solving systems of n equations using the bivariate spectral quasilinearization method (BSQLM) and bivariate spectral local linearization method (BSLLM). These methods, together with the bivariate spectral relaxation method (BSRM), were then used to solve a system of three nonlinear coupled partial differential equations that model an unsteady three-dimensional MHD-boundary-layer flow due to an impulsive motion of a stretching surface. Graphs and certain physical quantities were used to compare the accuracy and computational time of the methods. The BSLLM converged fastest, followed by the BSRM and then the BSQLM, as indicated by corresponding order of computational time. We can therefore conclude that the BSLLM is a better method for solving system of equations with $\eta \in [0, \infty)$ and $\zeta \in [0, 1]$.

Chapter 6

Legendre-Gauss-Lobbatto based bivariate pseudospectral quasilinearisation method for nonlinear evolution equations

This chapter presents a method termed the Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (LGL-BSQLM) for solving nonlinear evolution partial differential equations. The method employs quasilinearization and pseudospectral collocation with Lagrange interpolation polynomials and Legendre-Gauss-Lobbatto grid points. An error bound for the LGL-BSQLM is developed and proved in this chapter. The LGL-BSQLM is then used to find numerical solutions to the same six nonlinear evolution partial differential equations as were considered in Chapter 2. Exact analytical solutions from literature are compared with the numerical solutions to evaluate the accuracy, convergence and general performance of the LGL-BSQLM. To assess the overall performance of the proposed approach, the results are compared with the bivariate spectral quasilinearization method (BSQLM), which is based on Chebyshev-Gauss-Lobbatto grid points, as was introduced in Chapter 2. Performance of the two methods is compared in terms of accuracy, computational speed, condition number of coefficient matrices and convergence.

6.1 Introduction

As outlined earlier in Chapter 2, nonlinear partial differential equations (PDEs) are used to model many naturally occurring phenomena, but the nature of the equations presents challenges in finding explicit analytical solutions. Therefore, robust numerical methods that are computational fast, converge quickly and are accurate should be developed.

In Chapter 2, we made use of pseudospectral methods and showed that they gave accurate solutions of differential equations, using only a few grid points with minimal computational time for problems with smooth solutions. To solve nonlinear PDEs, some studies have combined spectral collocation methods with finite difference techniques. Spectral collocation with Lagrange interpolation approximates the space derivatives and the resulting nonlinear ODE is then solved using finite difference techniques. For example, Driscoll [66], Olmos [65], Javidi JAVIDI2005,JAVIDI2006 and Dehghan [40] solved the Fitzhugh-Nagumo, Fisher, Burgers-Fisher and Burgers-Huxley equations respectively using a combination of the Chebyshev spectral collocation method with Lagrange interpolation polynomials and the fourth-order Runge-Kutta method. To be specific, pseudospectral methods with Lagrange interpolation polynomials have been used to solve different problems arising from fluid mechanics [96, 141, 142].

In Chapter 2, we showed the development of a more accurate spectral collocation method with Lagrange interpolation polynomial called the bivariate spectral quasilinearization method (BSQLM), which was published recently as Motsa et al. [97]. This method uses quasilinearization techniques developed by Bellman and Kalaba [71] to linearize the nonlinear PDEs; that is using Chebyshev-Gauss-Lobbatto grid points with bivariate Lagrange interpolation. In that method, Chebyshev spectral collocation with Lagrange interpolation polynomial is applied independently in both space and time variables of the quasilinear partial differential equation. Out of curiosity, we wanted to experiment if changing the grid points will have an effect in the accuracy of the method, we decided to use different grid points and examined the accuracy of the method. In this chapter, we introduce an alternative approach that uses Legendre-Gauss-Lobbatto grid points with bivariate Lagrange interpolation instead of Chebyshev-Gauss-Lobbatto grid points with bivariate Lagrange interpolation.

The main objective of this chapter is to introduce the alternative method that uses spectral collocation, Legendre-Gauss-Lobbatto grid points with bivariate Lagrange interpolation polyno-

mials together with the quasilinearization method. To this end, nonlinear evolution equations are first linearized using the quasilinearization method. A pseudospectral collocation method with Lagrange interpolation polynomials is applied independently both in space and time variables to the quasilinearized evolution partial differential equation with Legendre-Gauss-Lobbatto grid points. We term this new approach, the Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (LGL-BSQLM). An error bound for the LGL-BSQLM is developed and proved in this chapter.

As we did in Chapter 2 with the BSQLM we test the application, accuracy and reliability of the proposed LGL-BSQLM by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and modified KdV equation The approximate solutions obtained using the LGL-BSQLM are compared against known explicit exact solutions from the literature, as well as the BSQLM, to assess its accuracy, computational speed and general performance. Tables show the maximum errors of the method for different numbers of spatial grid points, and the time taken to compute the approximate solutions. Convergence graphs and condition numbers are used to compare the two methods.

The remainder of this chapter is organized as follows. We present the general properties of Legendre polynomials in Section 6.2. Then in Section 6.3, we introduce the LGL-BSQLM algorithm for a general nonlinear evolution PDE. We present the error bounds in Section 6.4. In Section 6.5, we describe the application of the LGL-BSQLM to selected test problems. The numerical simulations and results are presented in Section 6.6, with our conclusions given in Section 6.7.

6.2 Legendre Polynomials

In this section, we briefly discuss the important properties of Legendre polynomials. We discuss their symmetric properties, orthogonality and the recurrence relation. Legendre polynomials are a special case of Jacobi polynomials [98, 138] $P_n^{(\alpha,\beta)}(x)$ with $\alpha = \beta = 0$. They are mutually orthogonal [98, 138, 143] with respect to the uniform weight function $\omega = 1$ and thus,

$$\int_{-1}^{1} P_n(x)P_k(x)dx = \frac{2\delta_{kn}}{2n+1}, \tag{6.1}$$

where

$$\delta_{kn} = \begin{cases} 0 & \text{if } k \neq n \\ 1 & \text{if } k = n. \end{cases} \tag{6.2}$$

The Legendre polynomial has the expansion

$$P_k(x) = \frac{1}{2^k} \sum_{l=0}^{[k/2]} (-1)^l \frac{(2k-2l)!}{2^k l!(k-l)!(k-2l)!} x^{k-2l} \tag{6.3}$$

where

$$[k/2] = \begin{cases} k/2 & \text{if } k \text{ is even} \\ (k-1)/2 & \text{if } k \text{ is odd.} \end{cases} \tag{6.4}$$

Equation (6.3), for $k \geq 0$, can be generated using Rodrigue's formula,

$$P_k(x) = \frac{1}{2^k k!} \frac{d^k}{dx^k} \left[(x^2-1)^k \right]. \tag{6.5}$$

Legendre polynomials are odd functions if the degree of the polynomial (k) is odd and even if k is even. Therefore, we have

$$P_k(-x) = (-1)^k P_k(x), \quad P_k(\pm x) = (\pm 1)^k. \tag{6.6}$$

Equation (6.6) is computed at the exact points x and hence reducing the effects of rounding errors. For $x \in [-1,1]$ and $k \geq 0$, Legendre polynomials have a uniform bound $|P_k(x)| \leq 1$. The recursive formula of Legendre polynomials [98, 138, 143] are given by:

$$(k+1)P_{k+1}(x) = (2k+1)xP_k(x) - kP_{k-1}(x), \text{ with } P_0(x) = 1, P_1(x) = x \tag{6.7}$$

for $k = 1, 2, \cdots, n$.

6.3 Bivariate Legendre Spectral Quasilinearization Method

In this section we introduce the Bivariate Legendre Spectral Quasilinearization Method (LGL-BSQLM) for approximating solutions of nonlinear evolution PDEs. For comparison purposes with the bivariate spectral quasilinearisation method based on the Lagrange interpolation poly-

nomial by Motsa et.al.[97], we consider nonlinear PDEs of the form,

$$\frac{\partial u}{\partial \xi} = \mathscr{H}\left(u, \frac{\partial u}{\partial \nu}, \frac{\partial^2 u}{\partial \nu^2}, \cdots, \frac{\partial^p u}{\partial \nu^p}\right), \tag{6.8}$$

in the physical region $\xi \in [0,T]$, $\nu \in [a,b]$. The order of differentiation is denoted by p, the required solution by $u(\nu,\xi)$ and $\mathscr{H}$ is the non-linear operator which contains all the spatial derivatives of u. The Legendre-Gauss-Lobbatto grid points and the corresponding differentiation matrices are defined in the interval $[-1,1]$. The time interval $\xi \in [0,T]$ and space region $\nu \in [a,b]$, are transformed to $t \in [-1,1]$ using the linear transformation $\xi = T(t+1)/2$ and $x \in [-1,1]$ using the linear transformation

$$\nu = \frac{1}{2}(b-a)x + \frac{1}{2}(b+a),$$

respectively. The general governing equation (6.8) can now be expressed as

$$\frac{\partial u}{\partial t} = H\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \cdots, \frac{\partial^n u}{\partial x^n}\right), \quad t \in [-1,1], \quad x \in [-1,1] \tag{6.9}$$

The assumption is that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u(x,t) \approx \sum_{i=0}^{N_x} \sum_{j=0}^{N_t} u(x_i,t_j)\mathscr{L}_i(x)\mathscr{L}_j(t). \tag{6.10}$$

The bivariate Lagrange interpolation polynomial interpolates $u(x,t)$ at selected points (x_i,t_j) in both the x and t directions, for $i = 0,1,2,\cdots,N_x$ and $j = 0,1,2,\cdots,N_t$. The grid points in time are given by the zeros of the polynomial

$$p(t) = (1-t^2)P'_{N_t}(t), \tag{6.11}$$

where

$$P_{N_t}(t) = \frac{1}{2^{N_t}} \sum_{k=0}^{[N_t/2]} \frac{(-1)^k (2N_t-2k)!\, t^{N_t-2k}}{2^{N_t} k!(N_t-k)!(N_t-2k)!} \tag{6.12}$$

is the Legendre polynomial. The first derivative of equation (6.12) is given by

$$P'_{N_t}(t) = \frac{N_t(N_t+1)}{(2N_t+1)}\left[\frac{P_{N_t-1}(t)-P_{N_t+1}(t)}{1-t^2}\right]. \tag{6.13}$$

The zeros of $P'_{N_t}(t)$ are computed using Newton's method. Hence given t_j^0 and for $k \geq 0$,$(1 \leq j \leq N_t-1)$, we have

$$t_j^{k+1} = t_j^k - \frac{P'_{N_t}(t_j^k)}{P''_{N_t}(t_j^k)}. \tag{6.14}$$

We use the relationship of the Sturm-Liouville problem to avoid evaluating the values of $P''_{N_t}(t)$. We note that

$$(1-t^2)P''_{N_t}(t) - 2tP'_{N_t}(t) + t(t+1)P_{N_t}(t) = 0, \tag{6.15}$$

and thus

$$P''_{N_t}(t) = \frac{2tP'_{N_t}(t) - t(t+1)P_{N_t}(t)}{(1-t^2)}. \tag{6.16}$$

Substituting equation (6.16) into equation (6.14), we get

$$t_j^{k+1} = t_j^k - \frac{(1-(t_j^k)^2)P'_{N_t}(t_j^k)}{2t_j^kP'_{N_t}(t_j^k) - t_j^k(t_j^k+1)P_{N_t}(t_j^k)}. \tag{6.17}$$

Similarly, the grid points in space are given by the zeros of the polynomial

$$p(x) = (1-x^2)P'_{N_x}(x), \tag{6.18}$$

and hence the zeros of $P'_{N_x}(x)$ can be obtained by using the modified Newton's method

$$x_j^{k+1} = x_j^k - \frac{(1-(x_j^k)^2)P'_{N_x}(x_j^k)}{2x_j^kP'_{N_x}(x_j^k) - x_j^k(x_j^k+1)P_{N_x}(x_j^k)}. \tag{6.19}$$

The zeros of equations (6.11) and (6.18) are called Legendre-Gauss-Lobatto points [144, 138, 145, 146]. The function $\mathscr{L}_i(x)$ is the characteristic Lagrange cardinal polynomial based on the Legendre-Gauss-Lobatto points [144, 138, 145, 146]

$$\mathscr{L}_i(x) = \prod_{\substack{i=0 \\ i\neq k}}^{N_x} \frac{x-x_k}{x_i-x_k}, \tag{6.20}$$

where

$$\mathscr{L}_i(x_k) = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k. \end{cases} \tag{6.21}$$

An alternative convenient expression for equation (6.20) is given by

$$\mathscr{L}_i(x) = \frac{1}{N_x(N_x+1)P_{N_x}(x_i)} \frac{(x^2-1)P'_{N_x}(x)}{(x-x_i)}. \tag{6.22}$$

We conveniently express equation (6.9) in the form:

$$H[u, u', \cdots, u^{(n)}] - \dot{u} = 0. \tag{6.23}$$

The dot and primes in equation (6.23) denote the time and space derivatives, respectively, and H is the nonlinear operator. We assume that the difference $u_{l+1} - u_l$ and all it's space derivatives are small, where l and $l+1$ denote previous and current iterations respectively. We approximate the nonlinear operator H using the linear terms of the Taylor series and hence

$$H[u, u', \cdots, u^{(n)}] \approx H[u_l, u'_l, \cdots, u_l^{(n)}] + \sum_{k=0}^{n} \frac{\partial H}{\partial u^{(k)}} \left(u_{l+1}^{(k)} - u_l^{(k)} \right) \tag{6.24}$$

Equation (6.24) can be expressed as

$$H[u, u', \cdots, u^{(n)}] \approx H[u_l, u'_l, \cdots, u_l^{(n)}] + \sum_{k=0}^{n} \zeta_{k,l}[u_l, u'_l, \cdots, u_l^{(n)}] u_{l+1}^{(k)} - \sum_{k=0}^{n} \zeta_{k,l}[u_l, u'_l, \cdots, u_l^{(n)}] u_l^{(k)} \tag{6.25}$$

where

$$\zeta_{k,l}[u_l, u'_l, \cdots, u_l^{(n)}] = \frac{\partial H}{\partial u^{(k)}}[u_l, u'_l, \cdots, u_l^{(n)}]. \tag{6.26}$$

Substituting equation (6.25) into equation (6.23), we get

$$\sum_{k=0}^{n} \zeta_{k,l} u_{l+1}^{(k)} - \dot{u}_{l+1} = \Omega_l[u_l, u'_l, \cdots, u_l^{(n)}] \tag{6.27}$$

where

$$\Omega_l[u_l, u_l', \cdots, u_l^{(n)}] = \sum_{k=0}^{n} \zeta_{k,l} u_l^{(k)} - H[u_l, u_l', \cdots, u_l^{(n)}]. \tag{6.28}$$

Collocation is an important step in the implementation of the solution procedure. Collocation in our case is the evaluation of the time derivative at the grid points t_j ($j = 0, 1, \ldots, N_t$) and the space derivatives at the grid points x_i ($i = 0, 1, \ldots, N_x$). We compute the values of the time derivatives at the Legendre-Gauss-Lobatto points (x_i, t_j), as (for $j = 0, 1, 2, \ldots, N_t$)

$$\left.\frac{\partial u}{\partial t}\right|_{(x_i,t_j)} = \sum_{s=0}^{N_x} \sum_{k=0}^{N_t} u(x_s, t_k) \mathscr{L}_s(x_i) \frac{d\mathscr{L}_k(t_j)}{dt} \tag{6.29}$$

$$= \sum_{k=0}^{N_t} u(x_i, t_k) d_{jk} = \sum_{k=0}^{N_t} d_{jk} u(x_i, t_k) \tag{6.30}$$

where $d_{jk} = \frac{d\mathscr{L}_k(t_j)}{dt}$ is the standard first derivative Legendre-Gauss-Lobatto based differentiation matrix of size $(N_t+1) \times (N_t+1)$ as defined in [138]. The first derivative matrix with respect to the Legendre-Gauss-Lobatto points is given by

$$d_{jk} = \begin{cases} -\frac{N_t(N_t+1)}{4} & \text{if } k = j = 0 \\ \frac{P_{N_t}(t_j)}{P_{N_t}(t_k)[t_j - t_k]} & \text{if } k \neq j, 0 \leq j, k \leq N_t \\ 0 & \text{if } 1 \leq j = k \leq N_t - 1 \\ \frac{N_t(N_t+1)}{4} & \text{if } k = j = N_t \end{cases} \tag{6.31}$$

We compute the values of the space derivatives at the Legendre-Gauss-Lobatto points (x_i, t_j) (for $i = 0, 1, 2, \cdots, N_x$) as

$$\left.\frac{\partial u}{\partial x}\right|_{(x_i,t_j)} = \sum_{s=0}^{N_x} \sum_{k=0}^{N_t} u(x_s, t_k) \frac{d\mathscr{L}_s(x_i)}{dx} \mathscr{L}_k(t_j) \tag{6.32}$$

$$= \sum_{s=0}^{N_x} u(x_s, t_j) D_{is} = \sum_{s=0}^{N_x} D_{is} u(x_s, t_j), \tag{6.33}$$

where $D_{is} = \frac{d\mathscr{L}_s(x_i)}{dx}$, is the standard first derivative Legendre-Gauss-Lobatto based differentiation matrix of size $(N_x+1)\times(N_x+1)$ as defined in [138]. The space first derivative matrix with respect to the Legendre-Gauss-Lobatto points is given by

$$D_{is} = \begin{cases} -\frac{N_t(N_t+1)}{4} & \text{if } s=i=0 \\ \frac{P_{N_t}(t_i)}{P_{N_t}(t_s)[t_i-t_s]} & \text{if } s\neq i, 0\leq i,s\leq N_t \\ 0 & \text{if } 1\leq i=s\leq N_t-1 \\ \frac{N_t(N_t+1)}{4} & \text{if } s=i=N_t \end{cases} \tag{6.34}$$

The space second derivative Legendre differentiation matrix with respect to the Legendre-Gauss-Lobatto points is given by [138]

$$D^2_{is} = \begin{cases} -2\frac{P_{N_x}(x_i)}{P_{N_x}(x_s)}\frac{1}{(x_i-x_s)^2} & \text{if } 1\leq i\leq N_x-1, 0\leq s\leq N_x, i\neq s \\ \frac{P''_{N_x}(x_s)}{3P_{N_x}(x_s)} & \text{if } 1\leq i=s\leq N_x-1 \\ \frac{(-1)^{N_x}}{P_{N_x}(x_s)}\frac{N_x(N_x+1)(1+x_s)-4}{2(1+x_s)^2} & \text{if } i=0, 1\leq s\leq N_x \\ \frac{1}{P_{N_x}(x_s)}\frac{N_x(N_x+1)(1-x_s)-4}{2(1-x_s)^2} & \text{if } i=N_x, 0\leq s\leq N_x-1 \\ \frac{N_x(N_x+1)(N_x^2+N_x-2)}{24} & \text{if } i=s=0, i=s=N_x \end{cases} \tag{6.35}$$

Similarly, the nth order derivative is defined as

$$\left.\frac{\partial^n u}{\partial x^n}\right|_{(x_i,t_j)} = \sum_{s=0}^{N_x} D^n_{is}u(x_s,t_j) = \mathbf{D}^n\mathbf{U}_j, \quad i=0,1,2,\ldots,N_x, \tag{6.36}$$

where the vector $\mathbf{U}_j$ is defined as

$$\mathbf{U}_j = [u(x_0,t_j), u(x_1,t_j), \ldots, u(x_{N_x},t_j)]^T. \tag{6.37}$$

and the superscript T denotes matrix transpose. Substituting (6.36) into (6.27) we get

$$\sum_{k=0}^{n} \mathbf{\Gamma}_{k,l}\mathbf{U}_{l+1,j}^{(k)} - \sum_{k=0}^{N_t} d_{jk}\mathbf{U}_{l+1,k} = \Omega_l[\mathbf{U}_{l,j}, \mathbf{U}'_{l,j}, \cdots, \mathbf{U}_{l,j}^{(n)}] \tag{6.38}$$

for $j = 0,1,2,\ldots,N_t$, where

$$\mathbf{U}_{l+1,j}^{(n)} = \mathbf{D}^n\mathbf{U}_{l+1,j}, \quad \mathbf{\Gamma}_{k,l} = \begin{bmatrix} \zeta_{k,l}(x_0,t_j) & & & \\ & \zeta_{k,l}(x_1,t_j) & & \\ & & \ddots & \\ & & & \zeta_{k,l}(x_{N_x},t_j) \end{bmatrix}. \tag{6.39}$$

Since the initial condition for equation (6.38) corresponds to $\xi_{N_t} = -1$, we express equation (6.38) as

$$\sum_{k=0}^{n} \mathbf{\Gamma}_{k,l}\mathbf{U}_{l+1,j}^{(k)} - \sum_{k=0}^{N_t-1} d_{jk}\mathbf{U}_{l+1,k} = \mathbf{\Omega}_j, \tag{6.40}$$

where

$$\mathbf{\Omega}_j = \Omega_l[\mathbf{U}_{l,j}, \mathbf{U}'_{l,j}, \cdots, \mathbf{U}_{l,j}^{(n)}] + d_{jN_t}\mathbf{U}_{N_t}, \quad j = 0,1,2,\ldots,N_t-1.$$

Equation (6.40) can be expressed as the following $N_t(N_x+1)\times N_t(N_x+1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{\Omega}_0 \\ \mathbf{\Omega}_1 \\ \vdots \\ \mathbf{\Omega}_{N_t-1} \end{bmatrix}, \tag{6.41}$$

where

$$A_{i,i} = \sum_{k=0}^{n} \mathbf{\Gamma}_{k,l} \mathbf{D}^{(k)} - d_{i,i}\mathbf{I} \tag{6.42}$$

$$A_{i,j} = -d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \tag{6.43}$$

and $\mathbf{I}$ is the identity matrix of size $(N_x+1)\times(N_x+1)$. Solving equation (6.40) gives $u(x_i,t_j)$ which is subsequently used in equation (6.10) to approximate $u(x,t)$.

6.4 Error Bounds

In this section, we present a theorem that can be used to quantify the error bound for the Legendre-Gauss-Lobbatto bivariate spectral quasilinearisation method for parabolic partial differential equations. We will first define certain concepts and a theorem and use these concepts to prove the theorem that will be presented shortly thereafter. Let us define the meaning of best approximation.

Definition 6.1. *Let $u(x,t)$ be a smooth function such that $(x,t) \in \Omega$, where $\Omega = [a,b]\times[0,T]$. Let $U(x,t) \in \Lambda_{N_x,N_t}$ denote the best approximation of $u(x,t)$, where*

$$\Lambda_{N_x,N_t} = span\left\{\mathscr{L}_i(x), \mathscr{L}_j(t), i = 0,1,\cdots,N_x, j = 0,1,\cdots,N_t\right\}. \tag{6.44}$$

$U_{N_x,N_t}(x,t) \in \Lambda_{N_x,N_t}$ being the best approximation of $u(x,t)$ means that there exist $v_{N_x,N_t}(x,t) \in \Lambda_{N_x,N_t}$ such that for all $v_{N_x,N_t}(x,t) \in \Lambda_{N_x,N_t}$, we have

$$|u(x,t) - U_{N_x,N_t}(x,t)| \leq |u(x,t) - v_{N_x,N_t}(x,t)|. \tag{6.45}$$

Theorem 6.1. *Let $u(x,t) \in C^{N_x+N_t+2}([a,b]\times[0,T])$ be sufficiently smooth such that there exist at least the (N_x+1) partial derivative with respect to x, (N_t+1) partial derivative with respect to t, and the (N_x+N_t+2) mixed partial derivatives with respect to x and t, and are all continuous.*

Then there exists some constants $\xi_x, \xi_x' \in (a,b)$, and $\xi_t, \xi_t' \in (0,T)$, such that

$$\begin{aligned} u(x,t)-U(x,t) = &\frac{\partial^{N_x+1}u(\xi_x,t)}{\partial x^{N_x+1}(N_x+1)!}\prod_{i=0}^{N_x}(x-x_i)+\frac{\partial^{N_t+1}u(x,\xi_t)}{\partial t^{N_t+1}(N_t+1)!}\prod_{j=0}^{N_t}(t-t_j) \\ &-\frac{\partial^{N_x+N_t+2}u(\xi_x',\xi_t')}{\partial x^{N_x+1}\partial t^{N_t+1}(N_x+1)!(N_t+1)!}\prod_{i=0}^{N_x}(x-x_i)\prod_{j=0}^{N_t}(t-t_j), \end{aligned} \tag{6.46}$$

where $U(x,t)$ is a polynomial interpolate of $u(x,t)$ at $(x_i)_{0\le i\le N_x}$ grid points in x-variable and $(t_j)_{0\le j\le N_t}$ grid points in t-variable.

Theorem 6.1 was first proposed by Gasca et.al [147]. Bhrawy et. al. [148] extended the theorem and used it to determine the error bound for general Jacobi polynomials. Bhrawy et. al. [148] obtained the error bound by taking the absolute value of equation (6.46) and observed that in general, the following hold:

$$\begin{aligned} |u(x,t)-U(x,t)| \le &\max_{(x,t)\in\Omega}\frac{1}{(N_x+1)!}\left|\frac{\partial^{N_x+1}u(\xi_x,t)}{\partial x^{N_x+1}}\right|\left|\prod_{i=0}^{N_x}(x-x_i)\right| \\ &+\max_{(x,t)\in\Omega}\frac{1}{(N_t+1)!}\left|\frac{\partial^{N_t+1}u(x,\xi_t)}{\partial t^{N_t+1}}\right|\left|\prod_{j=0}^{N_t}(t-t_j)\right| \\ &+\max_{(x,t)\in\Omega}\frac{1}{(N_x+1)!(N_t+1)!}\left|\frac{\partial^{N_x+N_t+2}u(\xi_x',\xi_t')}{\partial x^{N_x+1}\partial t^{N_t+1}}\right|\left|\prod_{i=0}^{N_x}(x-x_i)\right|\left|\prod_{j=0}^{N_t}(t-t_j)\right|. \end{aligned} \tag{6.47}$$

Using the fact that $u(x,t)$ is smooth on Ω, they concluded that its derivatives are bounded and hence there exists some constants α_1, α_2 and α_3, such that

$$\max_{(x,t)\in\Omega}\left|\frac{\partial^{N_x+1}u(x,t)}{\partial x^{N_x+1}}\right|\le\alpha_1,\quad \max_{(x,t)\in\Omega}\left|\frac{\partial^{N_t+1}u(x,t)}{\partial t^{N_t+1}}\right|\le\alpha_2,\quad \max_{(x,t)\in\Omega}\left|\frac{\partial^{N_x+N_t+2}u(x,t)}{\partial x^{N_x+1}\partial t^{N_t+1}}\right|\le\alpha_3. \tag{6.48}$$

Equation (6.48) implies that the error norm can be reduced by minimizing the expressions

$$\left|\prod_{i=0}^{N_x}(x-x_i)\right|,\quad \left|\prod_{j=0}^{N_t}(t-t_j)\right|. \tag{6.49}$$

A theorem that can be used to measure the error bound for the Legendre-Gauss-Lobbatto Bivariate spectral quasilinearisation method for parabolic partial differential equations is presented below.

Theorem 6.2. *Let $u(x,t) \in C^{N_x+N_t+2}([a,b]\times[0,T])$ be sufficiently smooth such that there exist at least the (N_x+1) partial derivative with respect to x, (N_t+1) partial derivative with respect to t, and the (N_x+N_t+2) mixed partial derivative with respect to x and t, and are all continuous. Then there exists some constants $\xi_x, \xi_x' \in (a,b)$, and $\xi_t, \xi_t' \in (0,T)$, such that*

$$\begin{aligned} u(x,t)-U(x,t) = \frac{\partial^{N_x+1}u(\xi_x,t)}{\partial x^{N_x+1}(N_x+1)!}\prod_{i=0}^{N_x}(x-x_i)+\frac{\partial^{N_t+1}u(x,\xi_t)}{\partial t^{N_t+1}(N_t+1)!}\prod_{j=0}^{N_t}(t-t_j) \\ -\frac{\partial^{N_x+N_t+2}u(\xi_x',\xi_t')}{\partial x^{N_x+1}\partial t^{N_t+1}(N_x+1)!(N_t+1)!}\prod_{i=0}^{N_x}(x-x_i)\prod_{j=0}^{N_t}(t-t_j), \end{aligned} \tag{6.50}$$

where $U(x,t)$ is a polynomial interpolant of $u(x,t)$ at the Legendre-Gauss-Lobbatto grid points $(x_i)_{0\le i\le N_x}$ in x-variable and Legendre-Gauss-Lobbatto grid points $(t_j)_{0\le j\le N_t}$ in t-variable. Then, the error bound is given by

$$\begin{aligned} |u(x,t)-U(x,t)| \le \alpha_1\frac{(b-a)^{N_x+1}}{2^{N_x+1}K_{N_x}(N_x+1)!}+\alpha_2\frac{(T)^{N_t+1}}{2^{N_t+1}K_{N_t}(N_t+1)!}+ \\ \alpha_3\frac{(b-a)^{N_x+1}(T)^{N_t+1}}{(2)^{(N_x+N_t+2)}K_{N_x}K_{N_t}(N_x+1)!(N_t+1)!}, \end{aligned} \tag{6.51}$$

where

$$K_{N_x} = \left(\frac{N_x}{N_x+1}\right)^2\left[\frac{(2N_x)!}{2^{N_x}(N_x!)^2}\right] \quad , \quad K_{N_t} \quad \textit{is similarly defined},$$

and α_1, α_2 and α_3 are constants defined by equation (6.48).

Proof: The Legendre-Gauss-Lobbatto grid points and the corresponding differentiation matrices are defined in the interval $[-1,1]$. Therefore, the time interval $t \in [0,T]$ and space region $x \in [a,b]$, are transformed to $\tau \in [-1,1]$ using the linear transformation

$$t = \frac{T}{2}(\tau+1) \tag{6.52}$$

and $\nu \in [-1,1]$ using the linear transformation

$$x = \frac{1}{2}(b-a)\nu + \frac{1}{2}(b+a), \tag{6.53}$$

respectively. The Legendre-Gauss-Lobbatto grid points in the τ direction are given by the zeros of the polynomial

$$p(\tau) = (1-\tau^2)P'_{N_t}(\tau), \tag{6.54}$$

and similarly, for the ν direction, they are given by the zeros of the polynomial

$$p(\nu) = (1-\nu^2)P'_{N_t}(\nu). \tag{6.55}$$

The Lagrange polynomial of order N_x+1 can be expressed in the form

$$L_{N_x+1}(\nu) = K_{N_x}\prod_{i=0}^{N_x}(\nu-\nu_i) \tag{6.56}$$

Equations (6.52) and (6.53) gives

$$x-x_i = \frac{(b-a)}{2}(\nu-\nu_i), \quad \text{and} \quad t-t_i = \frac{T}{2}(\tau-\tau_i) \tag{6.57}$$

Therefore, to minimize the error bound, we require that

$$\max_{a\le x\le b}\left|\prod_{i=0}^{N_x}(x-x_i)\right| = \max_{-1\le\nu\le 1}\left|\prod_{i=0}^{N_x}\frac{(b-a)}{2}(\nu-\nu_i)\right| \tag{6.58}$$

$$= \left(\frac{b-a}{2}\right)^{N_x+1}\max_{-1\le\nu\le 1}\left|\prod_{i=0}^{N_x}(\nu-\nu_i)\right| \tag{6.59}$$

$$= \left(\frac{b-a}{2}\right)^{N_x+1}\max_{-1\le\nu\le 1}\left|\frac{L_{N_x+1}(\nu)}{K_{N_x}}\right| \tag{6.60}$$

$$\le \left(\frac{b-a}{2}\right)^{N_x+1}\frac{1}{K_{N_x}} \quad \text{since} \quad |L_{N_x+1}(\nu)| \le 1 \tag{6.61}$$

Similarly, it can be shown that

$$\max_{0\le t\le T}\left|\prod_{j=0}^{N_t}(t-t_j)\right| = \left(\frac{T}{2}\right)^{N_t+1}\max_{-1\le\tau\le 1}\left|\frac{L_{N_t+1}(\tau)}{K_{N_t}}\right| \tag{6.62}$$

$$\le \left(\frac{T}{2}\right)^{N_t+1}\frac{1}{K_{N_t}}. \tag{6.63}$$

Applying equations (6.48), (6.61), and (6.63) into equation (6.47) yields the desired result. □

6.5 Numerical experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (6.9) with exact solutions. In order to determine the level of accuracy of the LGL-BSQLM approximate solution, at a particular time level, in comparison with the exact solution we report maximum error which is defined by

$$E_{N_x} = \max_i\left\{\left|u(x_i,t)-\tilde{u}(x_i,t)\right|, : 0\le i\le N_x\right\}, \tag{6.64}$$

where $\tilde{u}(x_i,t)$ is the approximate solution and $u(x_i,t)$ is the exact solution at the time level t. For the purpose of comparing with the BSQLM, we used the examples that were considered for numerical experiments by Motsa et. al. [97]. These numerical experiments are also listed in Chapter 2 of this thesis.

6.6 Results and Discussion

In this section, we discuss and present the results obtained by the new pseudospectral method, with bivariate Lagrange interpolation polynomial based on the Legendre-Gauss-Lobbatto points (LGL-BSQLM). We compare the results of this new spectral quasilinearization method with those obtained by quasilinearization using the bivariate Lagrange interpolation polynomial based on the Chebyshev-Gauss-Lobbatto points (BSQLM), as reported in Chapter 2. The results were all generated using MATLAB 2013. To compare the accuracy, computational time, and general performance of the proposed method, we compare the maximum errors and condition numbers

of the coefficient matrices obtained using both methods. We also show convergence graphs for both methods. We discuss the results in the following subsections.

Maximum Errors

In this subsection, we report the maximum errors of the nonlinear PDEs considered in this study. The maximum error is given by equation (6.64). In Tables 6.1 - 6.6, we used $N_t = 10$ and $4 \leq N_x \leq 6$ for comparison purposes.

Table 6.1 Maximum errors E_N for Fisher equation when α = 1 using $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	1.986e-008	1.119e-011	7.317e-009	5.763e-012
0.2	3.934e-008	3.121e-011	7.699e-009	6.269e-012
0.3	5.577e-008	4.864e-011	7.632e-009	6.041e-012
0.4	6.997e-008	6.802e-011	6.739e-009	4.776e-012
0.5	8.107e-008	7.971e-011	5.587e-009	3.815e-012
0.6	8.891e-008	8.560e-011	4.181e-009	2.547e-012
0.7	9.344e-008	8.953e-011	2.483e-009	7.282e-013
0.8	9.431e-008	8.759e-011	1.069e-009	1.450e-012
0.9	9.178e-008	8.325e-011	2.010e-009	2.681e-012
1.0	8.787e-008	7.421e-011	2.794e-009	3.480e-012
CPU Time (sec)	0.019942	0.025988	0.013729	0.014018

Table 6.1 compares the maximum errors for the Fisher equation (2.30) for $\alpha = 1$ obtained using the BSQLM and LGL-BSQLM. The LGL-BSQLM method converges to a smaller error compared to the BSQLM method. This implies that the LGL-BSQLM method is slightly accurate than the BSQLM method. We also note that both methods converge to small errors using very few grid points. This is reflected in the computational times of both methods. However, the LGL-BSQLM method converges faster than the BSQLM method as the method takes less time to converge to the exact solution. For $N_x = 4$, the BSQLM converges to $E_{N_x} \approx a \times 10^{-8}$ while

the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-9}$ for $a \geq b > 0$ for certain values of t. This implies that the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. The same trend is observed for $N_x = 6$.

Table 6.2 Maximum errors E_N for the Burgers-Fisher equation when α = 1 using $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	1.142e-007	1.369e-010	1.729e-008	2.278e-011
0.2	1.178e-007	1.373e-010	1.726e-008	2.325e-011
0.3	1.186e-007	1.479e-010	1.609e-008	1.702e-011
0.4	1.069e-007	9.450e-011	1.285e-008	1.210e-011
0.5	9.030e-008	7.944e-011	9.064e-009	6.378e-012
0.6	6.963e-008	6.618e-011	4.850e-009	1.159e-011
0.7	4.638e-008	1.579e-011	2.145e-009	1.163e-011
0.8	2.457e-008	4.030e-011	5.150e-009	1.233e-011
0.9	2.028e-008	6.006e-011	7.685e-009	1.488e-011
1.0	3.147e-008	7.708e-011	1.046e-008	1.458e-011
CPU Time (sec)	0.010152	0.015387	0.010061	0.010214

Table 6.2 compares the maximum errors for the Burgers-Fisher equation (2.15) for $\alpha = 1$ obtained using the BSQLM and LGL-BSQLM. In Table 6.2, the LGL-BSQLM method converges to a smaller error compared to the BSQLM method. We can therefore conclude that the LGL-BSQLM method is slightly accurate than the BSQLM method. We also note that both methods converge to small errors using very few grid points. This is reflected in the computational times of both methods. However, the LGL-BSQLM method converges faster than the BSQLM method as the method takes less time to converge to the exact solution. For $N_x = 4$, the BSQLM converges to $E_{N_x} \approx a \times 10^{-7}$ while the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-8}$ for $a \geq b > 0$ for certain values of t. This implies that the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. For $N_x = 6$, the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$.

Table 6.3 Maximum errors E_N for the Fitzhugh-Nagumo equation when α = 1 using $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	5.719e-007	1.196e-009	5.189e-008	1.239e-010
0.2	6.193e-007	1.299e-009	4.944e-008	1.045e-010
0.3	6.662e-007	1.463e-009	4.486e-008	9.320e-011
0.4	6.779e-007	1.448e-009	3.578e-008	7.368e-011
0.5	6.920e-007	1.526e-009	2.811e-008	5.933e-011
0.6	7.019e-007	1.573e-009	2.315e-008	4.363e-011
0.7	6.933e-007	1.516e-009	1.810e-008	2.299e-011
0.8	6.828e-007	1.535e-009	2.389e-008	3.746e-011
0.9	6.765e-007	1.528e-009	2.948e-008	5.536e-011
1.0	6.687e-007	1.490e-009	3.453e-008	7.131e-011
CPU Time (sec)	0.024281	0.024901	0.014825	0.015692

In Table 6.3, the maximum errors for the Fitzhugh-Nagumo equation (2.34) for $\alpha = 1$ obtained using the BSQLM method is compared with the LGL-BSQLM method. The LGL-BSQLM method converges to a smaller error compared to the BSQLM method. Therefore, the LGL-BSQLM method is slightly accurate than the BSQLM method. Both methods achieve accurate results using very few grid points and hence reducing the computational times of both methods. However, the LGL-BSQLM method converges faster than the BSQLM method as the method takes less time to converge to the exact solution. For $N_x = 4$, the BSQLM converges to $E_{N_x} \approx a \times 10^{-7}$ while the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-8}$ for $a \geq b > 0$ for some values of t. This implies that the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. For $N_x = 6$, the LGL-BSQLM method is also accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$.

Table 6.4 Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.75$, $\beta = 1$, $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	2.217e-006	8.482e-009	7.481e-007	2.888e-009
0.2	2.596e-006	9.369e-009	8.070e-007	2.895e-009
0.3	2.859e-006	1.073e-008	8.520e-007	3.078e-009
0.4	3.001e-006	1.112e-008	8.430e-007	3.015e-009
0.5	3.137e-006	1.213e-008	8.328e-007	3.007e-009
0.6	3.270e-006	1.311e-008	8.137e-007	2.952e-009
0.7	3.367e-006	1.359e-008	7.722e-007	2.762e-009
0.8	3.467e-006	1.438e-008	7.306e-007	2.590e-009
0.9	3.562e-006	1.504e-008	6.781e-007	2.341e-009
1.0	3.640e-006	1.559e-008	6.178e-007	2.057e-009
CPU Time (sec)	0.023822	0.024901	0.019958	0.022976

Table 6.4 compares the maximum errors for the Burgers-Huxley equation (2.38) for $\gamma = 0.75$, and $\beta = \delta = \alpha = 1$, obtained using the BSQLM and LGL-BSQLM. In Table 6.4, the LGL-BSQLM method converges to a smaller error compared to the BSQLM method and thus, the LGL-BSQLM method is slightly accurate than the BSQLM method. Spectral accuracy is achieved using few grid points and hence decreasing computational time. The LGL-BSQLM method converges faster than the BSQLM method as it takes less time to converge to the exact solution. The BSQLM converges to $E_{N_x} \approx a \times 10^{-6}$ while the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-7}$ for $N_x = 4$, $a \geq b > 0$ for certain values of t. Hence the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. Similarly, for $N_x = 6$, the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$.

Table 6.5 Maximum errors E_N for the modified KdV-Burgers equation when $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	1.80e-07	3.42e-10	2.21e-08	4.74e-11
0.2	2.61e-07	4.35e-10	2.35e-08	4.85e-11
0.3	2.72e-07	4.68e-10	2.49e-08	5.21e-11
0.4	2.01e-07	3.66e-10	2.21e-08	4.72e-11
0.5	2.58e-07	4.41e-10	2.42e-08	5.09e-11
0.6	2.65e-07	4.61e-10	2.45e-08	5.18e-11
0.7	2.25e-07	4.04e-10	2.31e-08	4.94e-11
0.8	2.57e-07	4.48e-10	2.46e-08	5.21e-11
0.9	2.44e-07	4.35e-10	2.37e-08	5.09e-11
1	8.28e-08	3.72e-10	1.49e-08	4.63e-11
CPU Time (sec)	0.015646	0.021226	0.013258	0.014976

Table 6.5 compares the maximum errors for the modified KdV-Burgers equation (2.43) obtained using the BSQLM and LGL-BSQLM. The modified KdV-Burgers equation (2.43) is a third order nonlinear PDE. In Table 6.5, the accuracy of the method does not deteriorate drastically with an increase in order and nonlinearity of the PDE. The LGL-BSQLM method converges to a smaller error compared to the BSQLM method and hence, the LGL-BSQLM method is slightly accurate than the BSQLM method. Spectral accuracy is achieved using few grid points and thus decreasing the computational time. The LGL-BSQLM method converges faster than the BSQLM method as it takes less time to converge to the exact solution. The BSQLM converges to $E_{N_x} \approx a \times 10^{-7}$ while the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-8}$ for $N_x = 4$, $a \geq b > 0$ for some values of t. Hence the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. Similarly, for $N_x = 6$, the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$.

Table 6.6 Maximum errors E_N for the modified KdV equation when $N_t = 10$

	Motsa [97] (BSQLM)		Current Results (LGL-BSQLM)	
$t \setminus N_x$	4	6	4	6
0.1	7.79e-05	3.55e-07	1.07e-05	5.11e-08
0.2	1.15e-04	4.00e-07	9.66e-06	4.30e-08
0.3	1.01e-04	3.74e-07	8.75e-06	3.96e-08
0.4	3.93e-05	1.79e-07	5.58e-06	2.65e-08
0.5	6.73e-05	2.34e-07	5.84e-06	2.59e-08
0.6	6.07e-05	2.21e-07	4.98e-06	2.26e-08
0.7	2.51e-05	1.11e-07	3.31e-06	1.56e-08
0.8	4.07e-05	1.43e-07	3.61e-06	1.60e-08
0.9	2.39e-05	1.02e-07	2.43e-06	1.17e-08
1	1.44e-04	7.26e-08	6.38e-06	4.82e-09
CPU Time (sec)	0.020609	0.021241	0.014958	0.020476

Table 6.6 compares the maximum errors for the modified KdV equation (2.47) obtained using the BSQLM and LGL-BSQLM. In Table 6.6, the accuracy of the method does not deteriorate drastically with an increase in order and non-linearity of the PDE. Table 6.6 depicts that the LGL-BSQLM method is slightly accurate than the BSQLM method. Spectral accuracy is achieved using few grid points and hence decreasing the computational time. The LGL-BSQLM method converges faster than the BSQLM method as it takes less time to converge to the exact solution. The BSQLM converges to $E_{N_x} \approx a \times 10^{-5}$ while the LGL-BSQLM converges to $E_{N_x} \approx b \times 10^{-6}$ for $N_x = 4$, $a \geq b > 0$ for some values of t. Hence the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$. Similarly, for $N_x = 6$, the LGL-BSQLM method is accurate by a factor of $A_{N_x} \approx |b-a| \times 10^{-1}$.

Condition Numbers

The condition number of a matrix plays a crucial role in numerical linear algebra for matrix computations. If the square matrix of the system $A\vec{x} = \vec{b}$ is non-singular, the condition number

becomes crucial in analyzing the convergence and accuracy of a method. The condition number of a square matrix A is defined by the equation

$$\kappa_p(A) = \|A\|_p \cdot \|A^{-1}\|_p. \tag{6.65}$$

The value of the condition number depends on the choice of the matrix norm and indirectly, of the vector norm. We can have $\kappa_\infty(A)$ or $\kappa_2(A)$ or $\kappa_1(A)$ for l_∞, l_2 and l_1 respectively. Since Lagrange and Legendre polynomials are both orthogonal polynomials in l_2 space, we used $\kappa_2(A)$ for the results in Tables 6.7 - 6.12. The condition number of the matrix is strongly connected to the accuracy of the underlying method used. It is always greater or equals to one. The square matrix A is said to *well conditioned* if the condition number is very small, otherwise it is *ill-conditioned*. If a matrix is *ill-conditioned*, then it can generate approximate solutions with a large error. The condition number of a matrix varies with different type of matrices.

In this section, we compare the condition numbers of the coefficient matrices of the BSQLM and LGL-BSQLM methods. We would expect that the condition number of a matrix will increase as the size of the matrix increases. However, the nature of the increase depends on the type of elements in the matrix. For all cases, when calculating the condition numbers, we used $N_t = 10$. We varied the size of the matrix by using different values of N_x and the condition numbers were captured at different iterations.

Table 6.7 Condition numbers $\kappa(A)$ for the Fisher's equation with α = 1 and $N_t = 10$.

	BSQLM			LGL-BSQLM		
Iterations $\setminus$ N_x	5	10	15	5	10	15
2	472.774	6059.973	35153.461	389.702	3833.080	20632.805
4	472.885	6061.612	35163.250	389.789	3834.073	20638.259
6	472.885	6061.612	35163.250	389.789	3834.073	20638.259
8	472.885	6061.612	35163.250	389.789	3834.073	20638.259

Table 6.8 Condition numbers $\kappa(A)$ for the Burgers-Fisher equation with $\alpha = \beta = \delta = 1$ and $N_t = 10$.

	BSQLM			LGL-BSQLM		
Iterations \ N_x	5	10	15	5	10	15
2	465.036	5942.506	34431.106	382.493	3752.184	20186.369
4	465.232	5945.532	34449.324	382.654	3754.044	20196.591
6	465.232	5945.532	34449.324	382.654	3754.044	20196.591
8	465.232	5945.532	34449.324	382.654	3754.044	20196.591

Table 6.9 Condition numbers $\kappa(A)$ for the Fitzhugh-Nagumo equation with $\alpha = 1$ and $N_t = 10$.

	BSQLM			LGL-BSQLM		
Iterations \ N_x	5	10	15	5	10	15
2	472.239	6053.600	35115.865	389.010	3826.584	20597.519
4	472.245	6053.699	35116.456	389.016	3826.643	20597.847
6	472.245	6053.699	35116.456	389.016	3826.643	20597.847
8	472.245	6053.699	35116.456	389.016	3826.643	20597.847

Table 6.10 Condition numbers $\kappa(A)$ for the Burgers-Huxley equation with $\gamma = 0.75$, $\alpha = \delta = \beta = 1$ and $N_t = 10$.

	BSQLM			LGL-BSQLM		
Iterations \ N_x	5	10	15	5	10	15
2	476.780	6123.607	35521.310	392.157	3867.183	20824.378
4	476.781	6123.618	35521.377	392.158	3867.190	20824.416
6	476.781	6123.618	35521.377	392.158	3867.190	20824.416
8	476.781	6123.618	35521.377	392.158	3867.190	20824.416

Table 6.11 Condition numbers $\kappa(A)$ for the modified KdV-Burgers equation with $N_t = 10$.

	BSQLM		LGL-BSQLM	
Iterations $\setminus N_x$	5	10	5	10
2	11766.897	971289.640	10124.969	695001.489
4	11766.897	971289.628	10124.969	695001.481
6	11766.897	971289.628	10124.969	695001.481
8	11766.897	971289.628	10124.969	695001.481

Table 6.12 Condition numbers $\kappa(A)$ for the modified KdV equation with $N_t = 10$.

	BSQLM		LGL-BSQLM	
Iterations $\setminus N_x$	5	10	5	10
2	11510.721	1114100.772	9985.449	799311.232
4	11497.159	1113689.776	9973.390	799013.536
6	11497.159	1113689.776	9973.390	799013.536
8	11497.159	1113689.776	9973.390	799013.536

The condition numbers of the matrices remain constant after two iterations in Tables 6.7 - 6.12. This is in agreement with the results of the convergent graphs in Figures 6.1 - 6.6. The condition numbers increase as the size of the matrix increases. In Tables 6.7 - 6.12, the condition numbers of the BSQLM method are larger than the condition numbers of the LGL-BSQLM method. This implies that the BSQLM method generate approximate solutions with large errors and hence less accurate than the LGL-BSQLM method. Matrices with small sizes are matrices with $N_x = 5$ and $N_t = 10$. For matrices with small sizes, both method's accuracy is comparable. However, as N_x increases, the condition numbers of the BSQLM method increases faster than that of the LGL-BSQLM and hence the BSQLM method becomes less accurate, as depicted in Tables 6.7 - 6.12.

Tables 6.7 - 6.10 present condition numbers for second order nonlinear parabolic PDEs. We note that for matrices with small sizes in Tables 6.7 - 6.10, the condition number is approximately

$$300 < \kappa_2(A) < 500. \tag{6.66}$$

However, for Tables 6.11 and 6.12, the condition numbers are for third order nonlinear parabolic PDEs. The condition numbers in Tables 6.11 and 6.12 for matrices with small sizes are approximately

$$9000 < \kappa_2(A) < 12000. \tag{6.67}$$

This implies that a matrix of a higher order nonlinear PDE will have a huge condition number and hence the method will generate approximate solutions with large errors. Thus the BSQLM method will generate less accurate results for higher order nonlinear PDEs compared to the LGL-BSQLM method.

Convergence Graphs

In this subsection, we compare the convergence and accuracy of the BSQLM and LGL-BSQLM methods graphically. In generating these figures, we used $N_x = 20$ and $N_t = 10$ with $t \in [0,1]$ and $x \in [0,1]$. In general, we observe that both methods converge almost after the same number of iterations. However, the LGL-BSQLM method converge to a smaller error than the BSQLM method. This implies that the LGL-BSQLM method is slightly accurate than the BSQLM method.

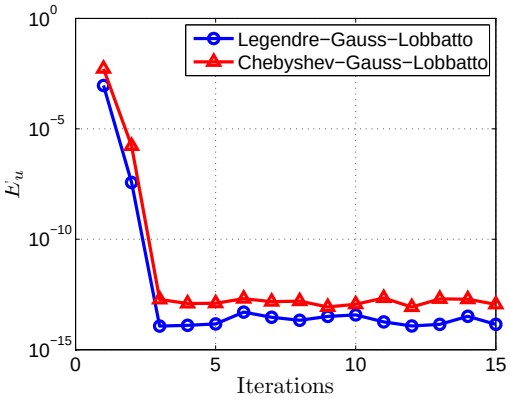

Fig. 6.1 Fisher's equation convergence graph

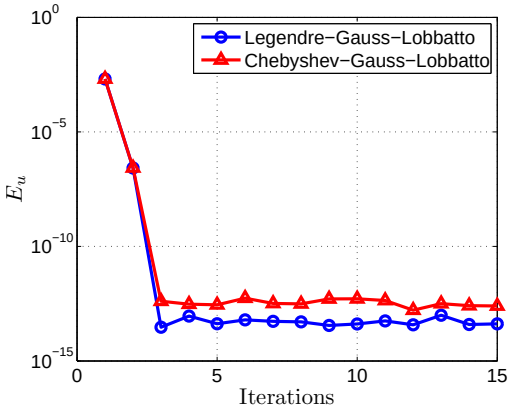

Fig. 6.2 Burgers-Fisher equation convergence graph

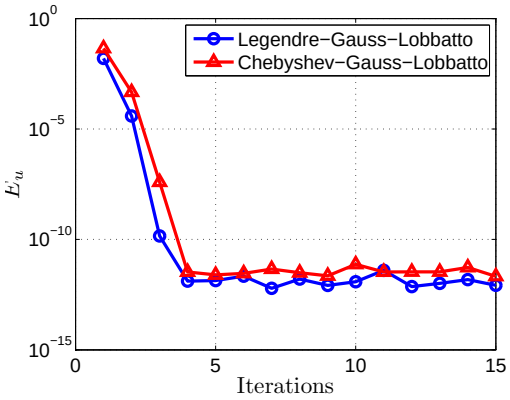

Fig. 6.3 Fitzhugh-Nagumo equation convergence graph

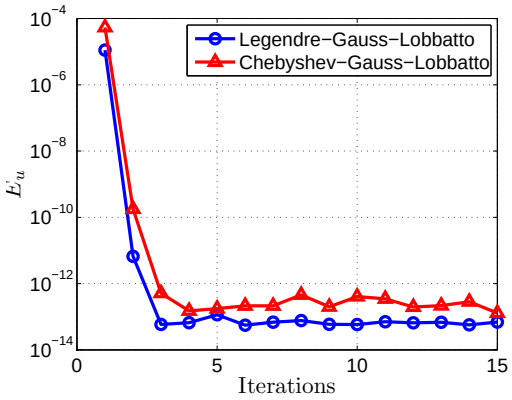

Fig. 6.4 Burgers-Huxley equation convergence graph

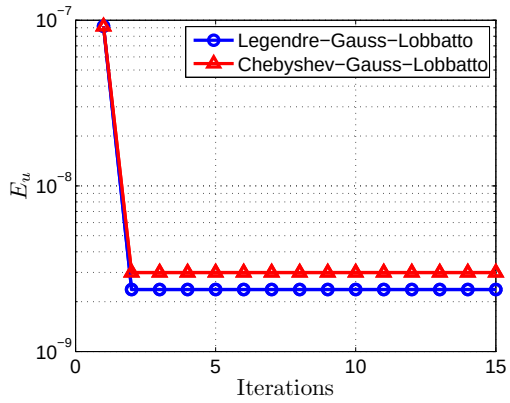

Fig. 6.5 Modified KdV-Burgers equation convergence graph

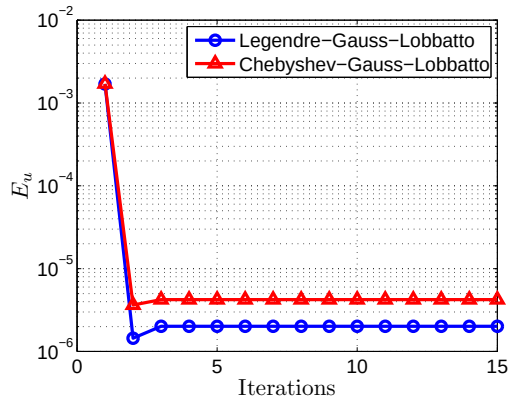

Fig. 6.6 Modified KdV equation convergence graph

6.7 Conclusion

In this chapter, a new approach termed the Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (LGL-BSQLM) for solving general nonlinear evolution smooth partial differential equations has been presented. The method was developed by quasilinearizing nonlinear partial differential equations and applying Lagrange interpolation polynomials in both space and time. The main goal in this chapter was to compare the accuracy, computational time and general performance of the LGL-BSQLM with the BSQLM from Chapter 2, in solving the same six nonlinear partial differential equations.

We carried out numerical simulations on the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and

modified KdV equation. We found that the LGL-BSQLM consistently converged to a slightly more accurate solution than did the BSQLM, as shown by the maximum error analysis, condition numbers and convergence graphs. The LGL-BSQLM also consistently used less computational time compared to the BSQLM. We can therefore conclude that the LGL-BSQLM algorithm gives slightly more accurate results and uses less computational time when compared to the BSQLM.

The work in this chapter makes a new contribution to the literature on quasilinearization techniques that can be used for solving nonlinear partial differential equations. Further research is needed to establish whether the LGL-BSQLM can be used successfully in solving coupled systems of equations arising from modeling other practical problems.

In the next chapter, we introduce another new approach for nonlinear partial differential equations over large time domains, which we call the multi-domain Legendre-Gauss-Lobatto based bivariate spectral quasilinearisation method (MD-LGL-BSQLM). It will be tested by solving the nonlinear evolution partial differential equations considered in Chapter 2 with large time domains.

Chapter 7

On the multi-domain bivariate quasilinearisation method for nonlinear evolution equations

This chapter presents a general approach for solving nonlinear evolution partial differential equations (NPDEs) over large time domains. The new approach combines the concepts of multi-domain, quasilinearization, spectral collocation and Lagrange interpolation polynomials with Legendre-Gauss-Lobatto grid points. In this chapter, strategies for implementing various boundary conditions is included. The new method is used to solve the same six nonlinear evolution partial differential equations used earlier in Chapters 2 and 6. The results are compared with results from the LGL-BSQLM technique from Chapter 6, and with known exact analytical solutions from literature in order to confirm accuracy, convergence and general performance of the new method.

7.1 Introduction

Evolution nonlinear partial differential equations (NPDEs) are useful tools for modeling naturally occurring phenomena but it is, however, difficult to obtain their analytical solutions due to their nonlinearity complexities over large time domains. Pseudospectral methods, such as were used in Chapters 2 and 6 have been shown to be computationally fast, converge quickly and be accurate. They use few grid points to achieve accurate solutions to differential equations, and

so they require minimal computational time. They have been successfully used to solve the equations in many scientific and engineering models.

As was noted in Chapters 2 and 6, some researchers (Li [149], Shamsi [144], and Ma [150]) have used spectral methods coupled with finite differences to solve partial differential equations. Spectral methods require few grid points to converge meanwhile finite difference methods require a lot of grid points to converge to an accurate solution. Therefore, the potential advantages of applying pseudospectral methods in space, is offset by the slower and less accurate finite differences in time.

In Chapter 2 (published as Motsa, Magagula, Sibanda [97]) we applied spectral collocation independently in both space and time in order to improve the accuracy and computational speed of pseudospectral methods. We showed that the BSQLM algorithm, which used Chebyshev spectral collocation method and bivariate Lagrange interpolation with Chebyshev-Gauss-Lobatto grid points, gives spectral accurate results, but only for small time domains. Then in Chapter 6, our numerical experiments showed that using Legendre-Gauss-Lobatto grid points leads to slightly more accurate solutions. Nevertheless, the method from Chapter 6 (LGL-BSQLM) gives spectral accurate results only for small time domains.

The main objective of this chapter is to introduce an alternative method that uses multi-domain, spectral collocation, bivariate Lagrange interpolation polynomials based on Legendre-Gauss-Lobatto grid points together with quasilinearization over large time domains. Accordingly, in this chapter, we now extend our work from Chapter 6 by incorporating the multi-domain approach in the t variable to increase the accuracy of the method for large values of t. Here we will use Legendre-Gauss-Lobatto grid points instead of Chebyshev-Gauss-Lobatto grid points in order to improve the accuracy of the proposed method. The multi-domain approach requires that the time domain be divided into smaller non-overlapping sub-intervals on which the pseudospectral collocation method is used to solve the partial differential equations. Continuity condition is then applied to advance the solution across the sub-intervals. The collocation based multi-domain approach had been used previously to solve systems of first order chaotic initial values problems [151–155].

We term this new approach the multi-domain Legendre-Gauss-Lobatto based bivariate Lagrange spectral quasilinearization method (MD-LGL-BSQLM). We test its applicability, accuracy and reliability by using it to solve the Fisher equation, Burgers-Fisher equation,

Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and nonlinear modified KdV equation. The results from the MD-LGL-BSQLM are then compared against known exact solutions that have been previously reported. We also conduct numerical experiments to evaluate the order of accuracy and convergence of the method and time taken to compute the solutions, and compare these with results using the LGL-BSQLM introduced in Chapter 6.

The organization of the chapter is as follows. In Section 7.2, we introduce the MD-LGL-BSQLM algorithm for a general nth order nonlinear evolution PDE. In Section 7.3 we describe the application of the MD-LGL-BSQLM using various boundary conditions. The numerical experiments and results are presented in Sections 7.4 and 7.5 respectively. Finally, our conclusions are given in Section 7.6.

7.2 Multi-domain Bivariate Lagrange Spectral Quasilinearization Method

In this section, the multi-domain based Legendre-Gauss-Lobbatto bivariate Lagrange spectral quasilinearization method (MD-LGL-BSQLM) for approximating solutions to nonlinear evolution partial differential equations is introduced. Without loss of generality, we consider an n^{th} order nonlinear partial differential equation of the form,

$$\frac{\partial u}{\partial \xi} = \mathscr{H}\left(u, \frac{\partial u}{\partial \nu}, \frac{\partial^2 u}{\partial \nu^2}, \cdots, \frac{\partial^n u}{\partial \nu^n}\right), \tag{7.1}$$

with the physical region $\xi \in [0,T]$, $\nu \in [a,b]$. The constant n denotes the order of differentiation, the required solution is denoted by $u(\nu,\xi)$ and $\mathscr{H}$ is the non-linear operator which contains $u(\nu,\xi)$ and all the spatial derivatives of $u(\nu,\xi)$. The multi-domain technique approach assumes that the time interval can be decomposed into p non-overlapping sub-intervals. Let $\xi \in \Gamma$ where $\Gamma = [0,T]$ be the time interval where the solution of general nonlinear parabolic PDE exist. The sub-intervals are defined as

$$\Gamma_l = (\xi_{l-1}, \xi_l), \quad l = 1,2,\cdots,p, \quad \text{with,} \quad 0 = \xi_0 < \xi_1 < \xi_2 < \cdots < \xi_p = T. \tag{7.2}$$

The main task in the multi-domain approach is determining the solution of equation (7.1) independently on each sub-interval, one at a time, beginning at the initial condition. The initial condition is considered to be the left boundary of the time interval. The computed solution in the first interval is used to compute the solutions in the remaining $l-1$ sub-intervals. The computed solution at the right hand boundary of the first interval is used as an initial condition in the subsequent sub-interval. The process is repeated until the last sub-interval. The process of matching the solutions in different intervals along their common boundary is called patching. The patching condition requires that

$$u^{(l)}(\nu,\xi_{l-1}) = u^{(l-1)}(\nu,\xi_{l-1}), \quad \nu \in [a,b], \tag{7.3}$$

where $u^{(l)}(\nu,\xi)$ denotes the solution of equation (7.1) at each sub-interval Γ_l. Since the grid points and differentiation matrices are defined in the interval $[-1,1]$, then, in each sub-interval Γ_l, the time interval, $\xi^l \in [\xi_{l-1},\xi_l]$ is transformed to $t \in [-1,1]$ using the linear transformation

$$\xi^l = \frac{1}{2}(\xi_l - \xi_{l-1})t + \frac{1}{2}(\xi_l + \xi_{l-1}) \tag{7.4}$$

Similarly, the space region, $\nu \in [a,b]$ is transformed using the linear transformation

$$\nu = \frac{1}{2}(b-a)x + \frac{1}{2}(b+a). \tag{7.5}$$

to the region $x \in [-1,1]$. Therefore, in the first sub-interval, we are required to solve the nonlinear parabolic equation

$$\frac{\partial u^{(1)}}{\partial t} = H\left(u^{(1)}, \frac{\partial u^{(1)}}{\partial x}, \frac{\partial^2 u^{(1)}}{\partial x^2}, \cdots, \frac{\partial^n u^{(1)}}{\partial x^n}\right), \quad t \in [-1,1], \quad x \in [-1,1]. \tag{7.6}$$

For $l \geq 2$, we solve the nonlinear parabolic equation

$$\frac{\partial u^{(l)}}{\partial t} = H\left(u^{(l)}, \frac{\partial u^{(l)}}{\partial x}, \frac{\partial^2 u^{(l)}}{\partial x^2}, \cdots, \frac{\partial^n u^{(l)}}{\partial x^n}\right), \quad t \in [-1,1], \quad x \in [-1,1]. \tag{7.7}$$

subject to

$$u^{(l)}(x,t_{l-1}) = u^{(l-1)}(x,t_{l-1}). \tag{7.8}$$

The method solution assumes that the solution at each sub-interval Γ_l, denoted by $u^{(l)}(x,t)$, can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u^{(l)}(x,t) \approx \sum_{i=0}^{M_x} \sum_{j=0}^{M_t} u^{(l)}(x_i,t_j)\mathscr{L}_i(x)\mathscr{L}_j(t). \tag{7.9}$$

The bivariate Lagrange interpolation polynomial interpolates $u^{(l)}(x,t)$ at carefully chosen grid points in both x and t directions called Legendre-Gauss-Lobatto points [98, 138]. Equation (7.6) can be expressed in the form:

$$H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \cdots, u^{(l)}_{(x,n)}\right] - u^{(l)}_{(t,1)} = 0, \tag{7.10}$$

where $u^{(l)}_{(x,n)}$ denotes the n^{th} partial derivative of $u(x,t)$ with respect to x in the l^{th} sub-interval. Similarly, $u^{(l)}_{(t,1)}$ denotes the 1^{st} partial derivative of $u(x,t)$ with respect to t in the l^{th} sub-interval and H is the nonlinear operator. We assume that the difference $u^{(l)}_{(x,0,s+1)} - u^{(l)}_{(x,0,s)}$ (note that s and $s+1$ denote previous and current iterations respectively.) and all it's space derivatives are small. The nonlinear operator H is approximated by using the linear terms of the Taylor series and thus

$$\begin{aligned} H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \cdots, u^{(l)}_{(x,n)}\right] \approx{}& H\left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right] \\ &+ \sum_{k=0}^{n} \frac{\partial H}{\partial u^{(l)}_{(x,k)}} \left(u^{(l)}_{(x,k,s+1)} - u^{(l)}_{(x,k,s)}\right). \end{aligned} \tag{7.11}$$

Let

$$\frac{\partial H}{\partial u^{(l)}_{(x,k)}} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right] = \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right]. \tag{7.12}$$

Therefore, equation (7.11) can be expressed as

$$\begin{aligned} H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \cdots, u^{(l)}_{(x,n)}\right] \approx{}& H\left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right] \\ &+ \sum_{k=0}^{n} \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right] u^{(l)}_{(x,k,s+1)} \\ &- \sum_{k=0}^{n} \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \cdots, u^{(l)}_{(x,n,s)}\right] u^{(l)}_{(x,k,s)} \end{aligned} \tag{7.13}$$

Let

$$R_s^{(l)}\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right]=\sum_{k=0}^{n}\omega_{k,s}^{(l)}\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right]u_{(x,k,s)}^{(l)}$$
$$-H\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right]. \tag{7.14}$$

Equation (7.13) can be expressed as

$$H\left[u_{(x,0)}^{(l)},u_{(x,1)}^{(l)},u_{(x,2)}^{(l)},\cdots,u_{(x,n)}^{(l)}\right]\approx\sum_{k=0}^{n}\omega_{k,s}^{(l)}\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right]u_{(x,k,s+1)}^{(l)}$$
$$-R_s^{(l)}\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right] \tag{7.15}$$

Substituting equation (7.15) into equation (7.10), we get

$$\sum_{k=0}^{n}\omega_{k,s}^{(l)}u_{(x,k,s+1)}^{(l)}-u_{(t,1,s+1)}^{(l)}=R_s^{(l)}\left[u_{(x,0,s)}^{(l)},u_{(x,1,s)}^{(l)},u_{(x,2,s)}^{(l)},\cdots,u_{(x,n,s)}^{(l)}\right] \tag{7.16}$$

Equation (7.16) is a linearized form of equation (7.6). The next non-trivial important procedure, termed collocation, is the evaluation of the time derivative at the Legendre-Gauss-Lobatto grid points t_j ($j=0,1,\ldots,M_t$) and the space derivatives at the Legendre-Gauss-Lobatto grid points x_i ($i=0,1,\ldots,M_x$). The values of the time derivatives are computed at the Legendre-Gauss-Lobatto points (x_i,t_j), as (for $j=0,1,2,\ldots,M_t$)

$$\left.\frac{\partial u^{(l)}}{\partial t}\right|_{(x_i,t_j)}=\sum_{\eta=0}^{M_t}d_{j\eta}u^{(l)}(x_i,t_\eta), \tag{7.17}$$

where for $l=1,2,\cdots,p$, $d_{j\eta}=\frac{d\mathscr{L}_\eta(t_j)}{dt}$, are the jth and ηth elements of the standard first derivative Legendre differentiation matrix of size $(M_t+1)\times(M_t+1)$. The nth order derivative is defined as

$$\left.\frac{\partial^n u^{(l)}}{\partial x^n}\right|_{(x_i,t_j)}=\sum_{\rho=0}^{M_x}D_{i\rho}^{n}u^{(l)}(x_\rho,t_j)=\mathbf{D}^n\mathbf{U}_j^{(l)},\quad i=0,1,2,\ldots,M_x, \tag{7.18}$$

where the vector $\mathbf{U}_j^{(l)}$ is defined as

$$\mathbf{U}_j^{(l)}=[u^{(l)}(x_0,t_j),u^{(l)}(x_1,t_j),\ldots,u^{(l)}(x_{M_x},t_j)]^T. \tag{7.19}$$

The superscript T in equation (7.19) denotes matrix transpose. Substituting equations (7.17) and (7.18) into equation (7.16), we get

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^k \mathbf{U}_{s+1,j}^{(l)} - \sum_{k=0}^{M_t} d_{jk} \mathbf{U}_{s+1,k}^{(l)} = R_s^{(l)} \tag{7.20}$$

for $j = 0, 1, 2, \ldots, M_t$, where $\mathbf{\Omega}_{k,r}$ is a diagonal matrix given by:

$$\mathbf{\Omega}_{k,r} = \begin{bmatrix} \omega_{k,s}(x_0, t_j) & & & \\ & \omega_{k,s}(x_1, t_j) & & \\ & & \ddots & \\ & & & \omega_{k,s}(x_{M_x}, t_j) \end{bmatrix}. \tag{7.21}$$

Since the initial condition for equation (7.20) corresponds to $\xi_{M_t} = -1$, we express equation (7.20) as

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^k \mathbf{U}_{s+1,j}^{(l)} - \sum_{k=0}^{M_t-1} d_{jk} \mathbf{U}_{s+1,k}^{(l)} = \mathbf{R}_j^{(l)}, \tag{7.22}$$

where

$$\mathbf{R}_j^{(l)} = R_s^{(l)} + d_{jM_t} \mathbf{U}_{M_t}^{(l)}, \quad \text{for} \quad j = 0, 1, 2, \ldots, M_t - 1.$$

For $j = 0, 1, 2, \ldots, M_t - 1$, equation (7.22) forms an $M_t(M_x+1) \times M_t(M_x+1)$ matrix equation

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,M_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,M_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M_t-1,0} & A_{M_t-1,1} & \cdots & A_{M_t-1,M_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0^{(l)} \\ \mathbf{U}_1^{(l)} \\ \vdots \\ \mathbf{U}_{M_t-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0^{(l)} \\ \mathbf{R}_1^{(l)} \\ \vdots \\ \mathbf{R}_{M_t-1}^{(l)} \end{bmatrix}, \tag{7.23}$$

where

$$A_{i,i} = \sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^{(k)} - d_{i,i} \mathbf{I} \tag{7.24}$$

$$A_{i,j} = -d_{i,j} \mathbf{I}, \quad \text{when } i \neq j, \tag{7.25}$$

and $\mathbf{I}$ is the identity matrix of size $(M_x+1)\times(M_x+1)$. Solving equation (7.23) gives $u(x_i,t_j)$ which is subsequently used in equation (7.9) to approximate $u^{(l)}(x,t)$.

7.3 Boundary Conditions

In this section, different type of boundary conditions are considered. We consider non-homogeneous Dirichlet boundary conditions, non-homogeneous Neumann boundary conditions and non-homogeneous mixed boundary boundary conditions.

Non-homogeneous Dirichlet boundary conditions

In this subsection, we consider boundary conditions of the form

$$u(a,t) = f_1(t) \tag{7.26}$$

$$u(b,t) = f_2(t) \tag{7.27}$$

in the interval $[a,b]$, where $f_1(t)$ and $f_2(t)$ are nonzero functions. In implementing non-homogeneous Dirichlet boundary conditions, we note that for $x \in [a,b]$, $x = a$ corresponds to $x = 1$, for $x \in [-1,1]$ and hence $u(a,t) = u(1,t)$. Similarly, for $x \in [a,b]$, $x = b$ corresponds to $x = 1$, for $x \in [-1,1]$ and hence $u(b,t) = u(-1,t)$. Therefore, when evaluating the boundary conditions at the Legendre-Gauss-Lobatto grid points in each sub-interval

$$\Gamma_l = (t_{l-1},t_l), \quad l = 1,2,\cdots,p, \quad \text{with,} \quad 0 = t_0 < t_1 < t_2 < \cdots < t_p = T, \tag{7.28}$$

we obtain

$$u^{(l)}(a,t_j) = u^{(l)}(x_{M_x},t_j) = f_1^{(l)}(t_j), \tag{7.29}$$

$$u^{(l)}(b,t_j) = u^{(l)}(x_0,t_j) = f_2^{(l)}(t_j), \tag{7.30}$$

for $j = 0,1,\ldots,M_t$.

Non-homogeneous Neumann boundary conditions

In this subsection, we consider boundary conditions of the form

$$u'(a,t) = g_1(t) \tag{7.31}$$

$$u'(b,t) = g_2(t) \tag{7.32}$$

in the interval $[a,b]$, where $g_1(t)$ and $g_2(t)$ are nonzero functions and the prime denotes differentiation with respect to x. In general, at any grid point (x_i,t_j), we have

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_i,t_j)} = \sum_{\rho=0}^{M_x} D_{i\rho} u^{(l)}(x_\rho,t_j). \tag{7.33}$$

Therefore, the non-homogeneous Neumann boundary conditions can be expressed as

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(a,t_j)} = \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_{M_x},t_j)} = \sum_{\rho=0}^{M_x} D_{M_x\rho} u^{(l)}(x_\rho,t_j) = g_1(t_j), \tag{7.34}$$

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(b,t_j)} = \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_0,t_j)} = \sum_{\rho=0}^{M_x} D_{0\rho} u^{(l)}(x_\rho,t_j) = g_2(t_j), \tag{7.35}$$

for $j = 0,1,\ldots,M_t$.

Non-homogeneous Mixed boundary conditions

In this subsection, we consider boundary conditions of the form

$$u(a,t)+u'(a,t) = h_1(t) \tag{7.36}$$

$$u(b,t)+u'(b,t) = h_2(t) \tag{7.37}$$

in the interval $[a,b]$, where $h_1(t)$ and $h_2(t)$ are nonzero functions and the prime denotes differentiation with respect to x. Using the ideas from the previous two subsections, equations (7.36)

and (7.37) can be expressed as

$$u^{(l)}(x_{M_x},t_j)+\sum_{\rho=0}^{M_x} D_{M_x\rho}u^{(l)}(x_\rho,t_j)=h_1(t_j), \tag{7.38}$$

$$u^{(l)}(x_0,t_j)+\sum_{\rho=0}^{M_x} D_{0\rho}u^{(l)}(x_\rho,t_j)=h_2(t_j), \tag{7.39}$$

for $j=0,1,\ldots,M_t$.

7.4 Numerical Experiments

In this section, we apply the proposed algorithm to the popular nonlinear partial differential equations of the form (7.6) with exact solutions. For comparison purposes, we considered the numerical experiments by Motsa et.al. [97]. These numerical experiments have been used in this thesis in Chapters 2 and 6. In all our calculations, we consider non-homogeneous Dirichlet boundary conditions. The space and time domains are given by $x\in[a,b]=[0,5]$ and $t\in[t_0,T]=[0,10]$ respectively for most of the numerical experiments. We choose a $T=10$ to show the accuracy of the MD-LGL-BSQLM over a large time domain.

7.5 Results and Discussion

In this section, we discuss and present the results of the MD-LGL-BSQLM method. The results were all generated using MATLAB 2013. To compare the accuracy, computational time, and general performance of the proposed method, we compare the maximum errors and convergence of the proposed method. In order to determine the level of accuracy of the MD-LGL-BSQLM approximate solution, at a particular time level in comparison with the exact solution, we report maximum error defined by

$$E_{M_x}=\max_k\left\{\left|u_e(x_k,t)-u_a(x_k,t)\right|,:0\le k\le M_x\right\}, \tag{7.40}$$

where $u_a(x_k,t)$ is the approximate solution and $u_e(x_k,t)$ is the exact solution at the time level t. We use $p=10$ and varying values of M_t and M_x.

Maximum Error estimates

In this subsection, we analyze the accuracy and computational time of the LGL-BSQLM and MD-LGL-BSQLM methods. We report the maximum errors obtained when solving the nonlinear evolution partial differential equations using both methods. We also report on the central processing unit (CPU) time taken to approximate the solutions of the nonlinear evolution partial differential equations. In all cases, we used $M_t = 10$ and varying values of M_x.

Table 7.1 Maximum error estimates E_{M_x} for the Fishers equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	1.840e-03	1.842e-03	9.508e-06	7.19e-10
2	2.906e-03	2.960e-03	1.033e-05	6.82e-10
3	1.150e-03	1.163e-03	4.555e-06	2.86e-10
4	7.918e-04	7.918e-04	2.597e-06	5.21e-11
5	5.383e-04	5.389e-04	5.600e-07	3.77e-12
6	3.721e-04	3.721e-04	3.176e-07	2.23e-12
7	2.494e-04	2.494e-04	6.682e-08	1.37e-13
8	1.172e-04	1.206e-04	1.077e-08	5.40e-14
9	1.264e-04	1.268e-04	9.716e-09	1.80e-14
10	2.210e-05	2.483e-05	6.086e-09	4.44e-15
CPU Time	0.015507	0.044061	0.005363	0.009849

In Table 7.1, the maximum error estimates for the Fisher equation are displayed. We observe that the maximum errors obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. This suggests that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve spectral accurate results. On average, for the same values of M_x and M_t, the difference between maximum error estimates obtained using the

MD-LGL-BSQLM and LGL-BSQLM method is of order ten. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.1.

Table 7.2 Maximum error estimates E_{M_x} for the Burgers-Fisher equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	1.779e-03	1.772e-03	1.006e-05	3.68e-09
2	2.054e-03	2.066e-03	1.570e-05	7.78e-10
3	2.977e-03	2.977e-03	8.765e-06	4.55e-11
4	2.366e-03	2.366e-03	1.343e-06	3.14e-11
5	5.866e-04	5.856e-04	4.678e-07	4.31e-12
6	1.392e-03	1.392e-03	2.175e-07	5.53e-13
7	1.135e-03	1.135e-03	7.135e-08	1.37e-14
8	4.054e-04	4.054e-04	2.135e-08	1.20e-14
9	1.920e-04	1.923e-04	6.240e-09	4.66e-15
10	9.550e-05	1.034e-04	1.803e-09	4.22e-15
CPU Time	0.025428	0.066148	0.011777	0.024576

Table 7.2, shows the maximum error estimates for the Burgers-Fisher equation. The maximum error estimates obtained using the MD-LGL-BSQLM method are smaller compared to those obtained using the LGL-BSQLM method. Therefore, we conclude that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses fewer grid points to achieve spectral accurate results. Increasing $M_x = 10$ gives an error of approximately 10^{-12} in the time domain $[0, 10]$. Increasing the number of space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.2. Solutions of the Burgers-Fisher equation were obtained in a fraction of a second.

Table 7.3 Maximum error estimates E_{M_x} for the Fitzhugh-Nagumo equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	1.770e-04	1.770e-04	6.049e-07	5.978e-11
2	2.383e-04	2.383e-04	6.071e-08	4.030e-13
3	4.189e-04	4.189e-04	1.037e-08	1.664e-14
4	3.305e-04	3.305e-04	2.430e-09	6.710e-15
5	1.480e-05	1.481e-05	7.108e-10	2.942e-15
6	2.413e-04	2.413e-04	2.455e-10	3.842e-15
7	2.198e-04	2.198e-04	9.683e-11	2.179e-15
8	8.642e-05	8.642e-05	4.264e-11	1.188e-15
9	4.103e-05	4.103e-05	2.056e-11	3.201e-16
10	1.595e-05	1.593e-05	1.065e-11	4.352e-16
CPU Time	0.064019	0.076709	0.013449	0.026672

Table 7.3, shows the maximum error estimates for the Fitzhugh-Nagumo equation, obtained using both the MD-LGL-BSQLM and LGL-BSQLM methods. The maximum errors obtained using the MD-LGL-BSQLM method are smaller compared to those obtained using the LGL-BSQLM method. Therefore, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. Using $M_x = 5$ reduces the size of the variable matrix and in turn reducing computational time. Thus the MD-LGL-BSQLM method uses fewer grid points to achieve more accurate results for large time domains. Increasing $M_x = 10$ gives an error of approximately 10^{-14} in the time domain $[0, 10]$. Increasing the number of space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.3. Solutions of the Fitzhugh-Nagumo equation were obtained using minimal computational time.

Table 7.4 Maximum error estimates E_{M_x} for the Burgers-Huxley equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	8.727e-04	8.705e-04	6.245e-17	1.735e-17
2	1.943e-03	1.950e-03	4.580e-16	9.021e-17
3	1.746e-03	1.773e-03	5.135e-16	4.857e-16
4	1.820e-03	1.820e-03	1.110e-15	9.437e-16
5	8.588e-04	8.507e-04	4.441e-15	4.441e-16
6	1.824e-03	1.824e-03	8.882e-16	3.109e-15
7	1.479e-03	1.479e-03	6.439e-15	1.332e-15
8	4.790e-04	4.790e-04	2.220e-15	4.885e-15
9	2.396e-04	2.556e-04	1.288e-14	3.997e-15
10	1.586e-04	1.599e-04	3.220e-15	6.661e-16
CPU Time	0.040081	0.084866	0.013389	0.019958

Table 7.4 displays the maximum error estimates for the Burgers-Huxley equation which were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. It is evident from Table 7.4 that the maximum errors obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. Thus the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. For $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. On average, for the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately twelve. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.4. The solutions of the Burgers-Huxley equation were also obtained in a fraction of a second.

Table 7.5 Maximum error estimates E_{M_x} for the KdV-Burgers equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	3.190e-04	3.182e-04	9.086e-07	2.114e-10
2	2.913e-04	2.918e-04	4.190e-07	4.539e-11
3	6.550e-04	6.550e-04	2.285e-07	1.660e-11
4	6.249e-04	6.249e-04	1.097e-07	6.400e-12
5	1.319e-04	1.320e-04	5.032e-08	2.561e-12
6	4.568e-04	4.568e-04	2.284e-08	1.174e-12
7	3.446e-04	3.446e-04	1.038e-08	5.967e-13
8	6.861e-05	7.126e-05	4.743e-09	3.062e-13
9	5.261e-05	5.266e-05	2.185e-09	1.230e-13
10	4.016e-05	4.714e-05	1.016e-09	5.757e-14

Table 7.5 shows the maximum error estimates for the KdV-Burgers equation which were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. In this table, the maximum error estimates obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. Thus the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We observe that for $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. For the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately eleven. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.5. The solutions of the Burgers-Huxley equation were also obtained in a fraction of a second.

Table 7.6 Maximum error estimates E_{M_x} for the modified KdV equation, with $M_t = 10$

	LGL-BSQLM		MD-LGL-BSQLM	
$t \setminus M_x$	10	20	5	10
1	4.600e-04	4.794e-04	1.278e-06	4.53e-10
2	2.859e-04	2.859e-04	6.520e-07	1.30e-10
3	5.044e-04	5.044e-04	3.413e-07	4.66e-11
4	3.985e-04	3.985e-04	1.641e-07	1.76e-11
5	8.550e-05	8.785e-05	7.270e-08	7.09e-12
6	2.911e-04	2.911e-04	3.223e-08	2.96e-12
7	2.651e-04	2.651e-04	1.490e-08	1.34e-12
8	1.042e-04	1.042e-04	6.970e-09	5.31e-13
9	4.946e-05	4.946e-05	3.183e-09	2.72e-13
10	5.541e-05	5.556e-05	1.455e-09	5.01e-13
CPU Times	0.023869	0.087991	0.006778	0.011656

Table 7.6 shows the maximum error estimates for the modified KdV equation that were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. The maximum error estimates obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. Therefore, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We observe that for $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. For the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately twelve. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 7.6. The solutions of the Burgers-Huxley equation were also obtained in a fraction of a second.

In summary, the proposed method achieves spectrally accurate results with relatively fewer spatial grid points, converges fast to the exact solution. It also approximates the solution of the

problem in minimal computational time. Solutions of nonlinear partial differential equations are obtained in few seconds in all cases.

Comparison of Approximate and Exact solutions

In this subsection, we consider the approximate and analytical solutions of the nonlinear partial differential equations considered in this work. The graphs show the solutions of the nonlinear partial differential equations in both space and time domains. Figures 7.1 - 7.6 show the approximate and exact solutions of Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation, Fitzhugh-Nagumo equation, modified KdV-Burgers equation and modified KdV equation respectively. These graphs were generated using the MD-LGL-BSQLM method. The graphs show that the approximate and exact solutions are in excellent agreement in the given time domain for all the equations considered. This implies that the MD-LGL-BSQLM method can be used to approximate solutions of nonlinear partial differential equations in large time domains. These graphs were generated using $M_x = 40$ and $M_t = 10$. The time and space intervals used are $t \in [0, 10]$ and $x \in [0, 5]$ for all the equations.

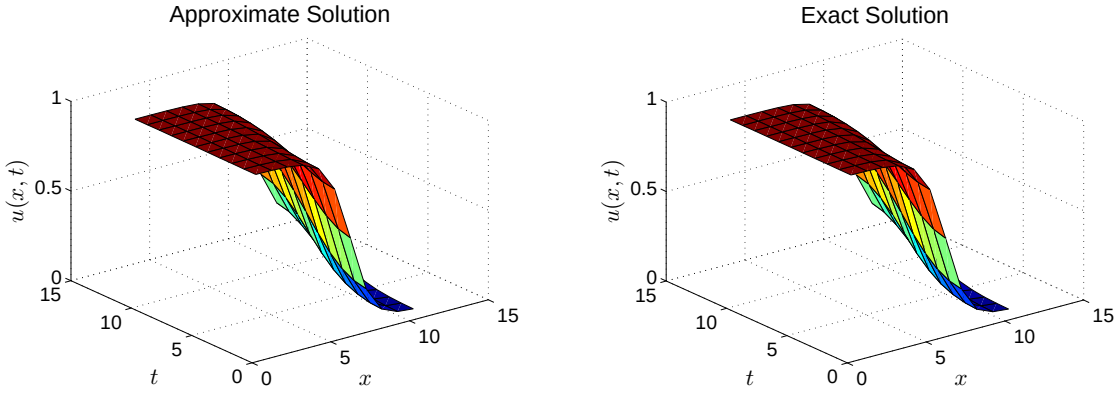

Fig. 7.1 Approximate and Exact solutions of the Fishers equation.

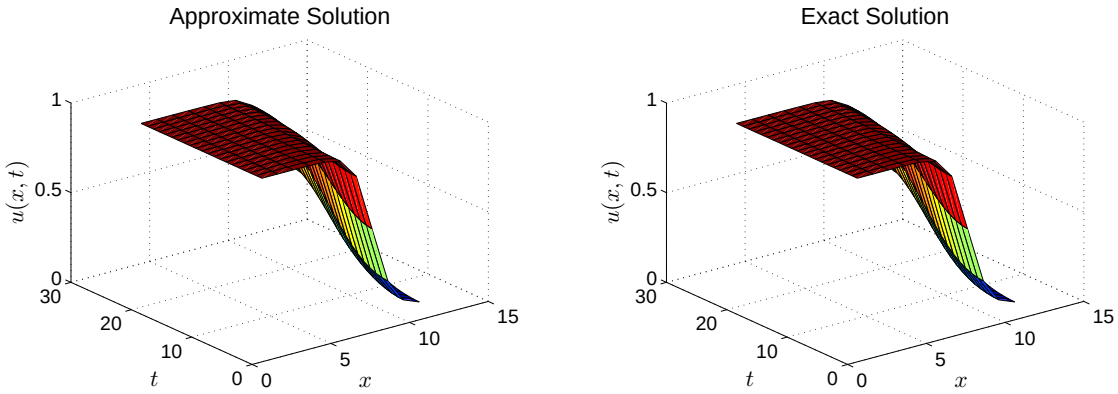

Fig. 7.2 Approximate and Exact solutions of the Burgers-Fisher equation.

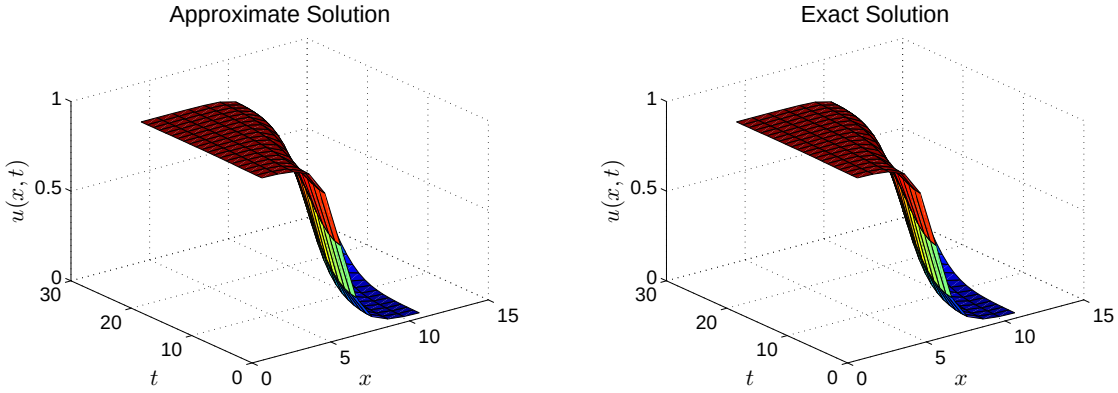

Fig. 7.3 Approximate and Exact solutions of the Burgers-Huxley equation.

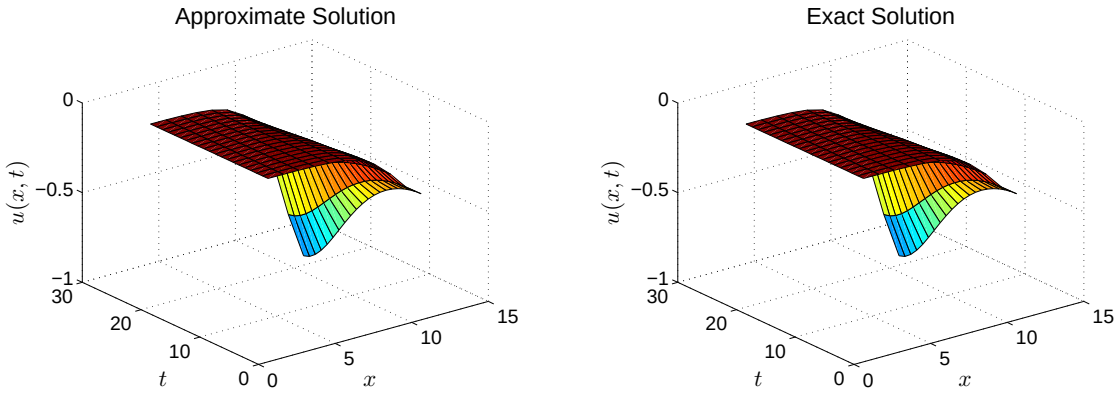

Fig. 7.4 Approximate and Exact solutions of the Fitzhugh-Nagumo equation.

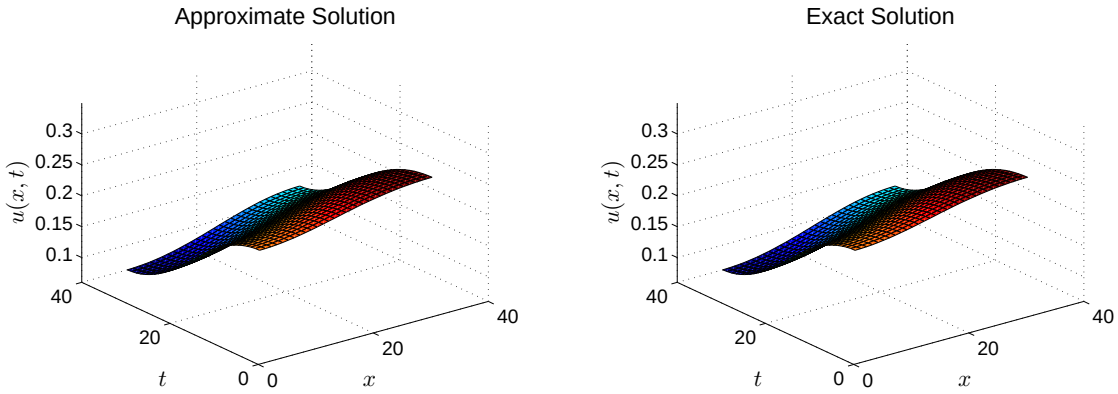

Fig. 7.5 Approximate and Exact solutions of the modified KdV-Burgers equation.

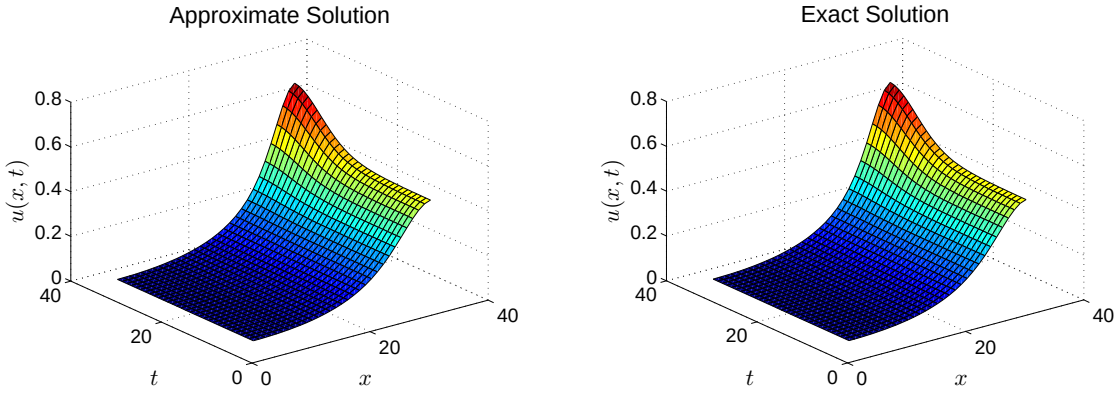

Fig. 7.6 Approximate and Exact solutions of the modified KdV equation.

Convergence Graphs

In this section, we compare the convergence and accuracy of the LGL-BSQLM and MD-LGL-BSQLM methods graphically. In generating the results, we used $M_x = M_t = 10$, $t \in [0,10]$, and $x \in [0,5]$. In general, the MD-LGL-BSQLM method converge approximately after two iterations, to a smaller error compared to the LGL-BSQLM method. This implies that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method.

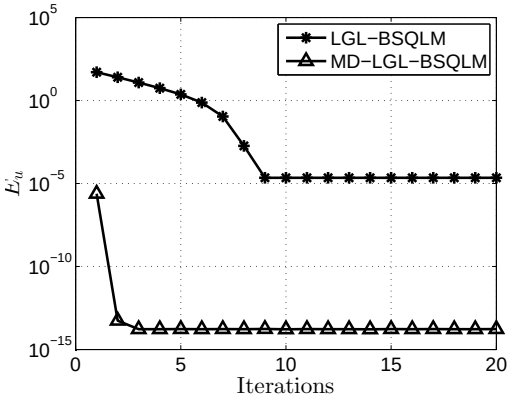

Fig. 7.7 Fisher's equation convergence graph

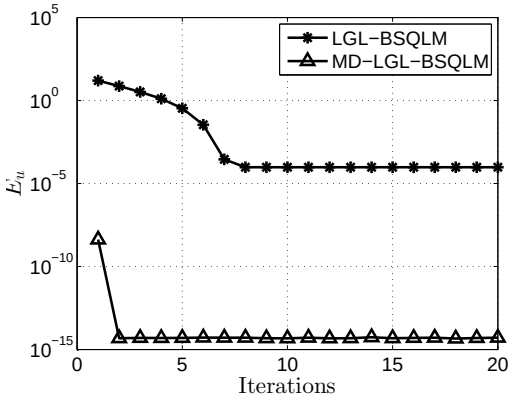

Fig. 7.8 Burgers-Fisher equation convergence graph

Figure 7.7 compares the convergence of the Fisher's equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. The MD-LGL-BSQLM method converges approximately after two iterations to an error of 10^{-14}, while the LGL-BSQLM method converges to 10^{-5} after nine iterations. This implies that the MD-LGL-BSQLM method is a suitable method to approximate the solution of the Fisher's equation for large time domains since it gives more accurate results and converges faster to the exact solution compared to the LGL-BSQLM method.

Figure 7.8 compares the convergence of the Burgers-Fisher equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. In this case, the MD-LGL-BSQLM method converges to 10^{-12} approximately after two iterations, while the LGL-BSQLM method converges to 10^{-4} after seven iterations. Since the MD-LGL-BSQLM method gives more accurate results and converges faster to the exact solution than the LGL-BSQLM method, then it is a suitable method to approximate the solution of the Burgers-Fisher equation for large time domains.

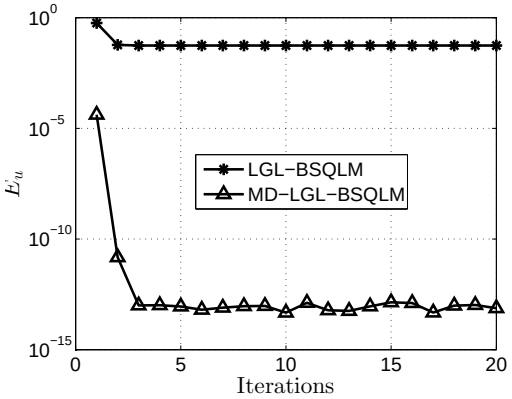

Fig. 7.9 Fitzhugh-Nagumo equation convergence graph

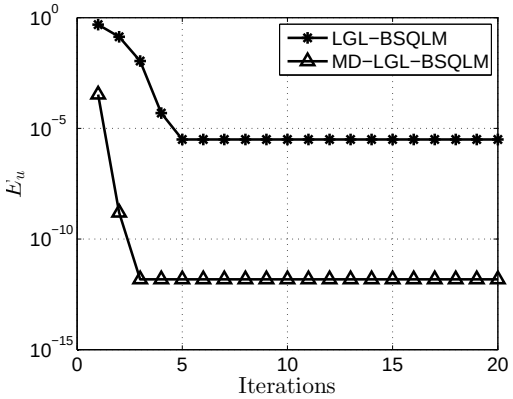

Fig. 7.10 Burgers-Huxley equation convergence graph

Figure 7.9 compares the convergence of the Fitzhugh-Nagumo equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. We observe that the MD-LGL-BSQLM method converges to 10^{-13} after three iterations, while the LGL-BSQLM method converges to 10^{-1} after two iterations. The LGL-BSQLM method converges to a less accurate solution compared to the MD-LGL-BSQLM method. This implies that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method and hence suitable for approximating the solution of the Fitzhugh-Nagumo equation for large time intervals.

In Figure 7.10, the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of the Burgers-Huxley equation is compared. The graph shows that the MD-LGL-BSQLM method converges to 10^{-12} after two iterations, while the LGL-BSQLM method converges to 10^{-6} after five iterations. Clearly, the LGL-BSQLM method converges to a less accurate solution compared to the MD-LGL-BSQLM method. Thus the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method and hence it is suitable for approximating the solution of the Fitzhugh-Nagumo equation for large time intervals.

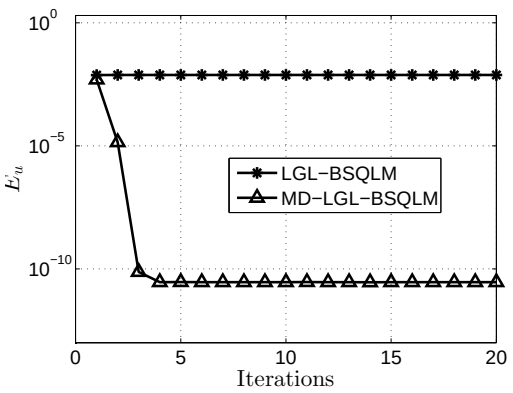

Fig. 7.11 Modified KdV-Burgers equation convergence graph

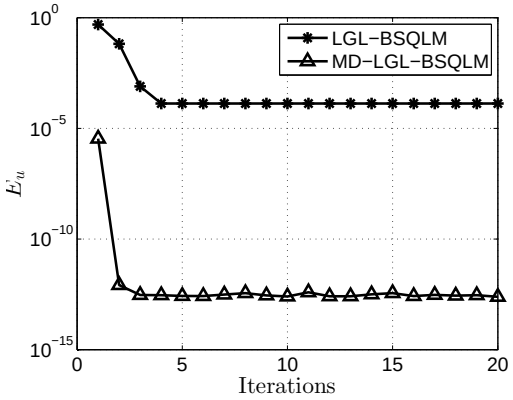

Fig. 7.12 Modified KdV equation convergence graph

Figure 7.11 compares the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of modified KdV-Burgers equation. The MD-LGL-BSQLM method converges to 10^{-11} after four iterations, while the LGL-BSQLM method converges to 10^{-2} after two iterations. We observe that the LGL-BSQLM method converges to a less accurate solution compared to the MD-LGL-BSQLM method and hence the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. This implies that the MD-LGL-BSQLM

method is suitable for approximating solutions of the modified KdV-Burgers equation for large time intervals.

Lastly, Figure 7.12 compares the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of modified KdV equation. The MD-LGL-BSQLM method converges to 10^{-13} after two iterations, while the LGL-BSQLM method converges to 10^{-4} after four iterations. We observe that the MD-LGL-BSQLM method converges to a more accurate solution compared to the LGL-BSQLM method and hence the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. Thus the MD-LGL-BSQLM method is suitable for approximating solutions of the modified KdV equation for large time intervals.

In summary, the convergence graphs show that the MD-LGL-BSQLM method converges to a more accurate solution than the LGL-BSQLM method. We can conclude that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method for solving nonlinear evolution partial differential equations in large time domains.

7.6 Conclusion

The main goal of this chapter was to develop a new numerical method for solving nonlinear partial differential equations over a long period of time. Accordingly, we presented and implemented for the first time a new approach termed the multi-domain Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method for solving general nonlinear evolution smooth partial differential equations. The proposed method was developed by quasilinearizing the evolution equations and independently applying Lagrange interpolation polynomials based on Legendre-Gauss-Lobatto grid points in both space and time. The time domain was further divided into small sub-intervals and the linearized evolution equations were solved in each sub-interval.

Numerical simulations using this new method were carried out on the Fisher equation, Burgers-Fisher equation, Burgers-Huxley equation, Fitzhugh-Nagumo equation, modified KdV-Burgers equation and modified KdV equation. The results indicate agreement between the approximate solutions and the analytical solutions to a high order of accuracy. Furthermore, over large time domains the MD-LGL-BSQLM converged to a more spectrally accurate solution than did the LGL-BSQLM. The MD-LGL-BSQLM also converged more quickly and used minimal

computational time compared to the LGL-BSQLM. We therefore conclude that the multi-domain Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method performs much better, in terms of both accuracy and computational speed, than the Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method.

This study contributes to the literature on new quasilinearization techniques that can be used for solving nonlinear partial differential equations over large time domains. In the next chapter, we extend the idea of using the multi-domain domain approach coupled with the bivariate pseudospectral methods (from Chapters 3, 4 and 5) to solve system of n equations.

Chapter 8

Multi-domain bivariate spectral collocation methods for systems of non-similar boundary layer differential equations

In this chapter, novel approaches for solving systems of non-similar boundary layer equations over a large time domain are presented. The methods are a multi-domain bivariate spectral quasilinearization method (MD-BSQLM) and multi-domain bivariate spectral local linearization method (MD-BSLLM). These methods make use of Legendre-Gauss-Lobbatto grid points, a linearization technique, and the spectral collocation method to approximate functions defined by bivariate Lagrange interpolation. The methods are developed for a general system of n nonlinear partial differential equations.

We demonstrate the use of the MD-BSQLM and MD-BSLLM techniques by solving a system of nonlinear partial differential equations that describe a class of non-similar boundary layer equations. Numerical experiments are conducted to show applicability and accuracy of the methods. Grid independence tests establish their accuracy, convergence and validity. The solution for the limiting case is used to validate their accuracy.

8.1 Introduction

As was outlined in earlier chapters, solutions of nonlinear non-similar boundary layer partial differential equations have received considerable attention of research scientist in the past few years. Series solution methods have been used by several researchers. These include asymptotic methods [156, 157], extended series solutions [157] and homotopy analysis method [158, 159]. However, solving a complex non-similar boundary layer differential equation by series methods may require many terms which infers extensive computational time.

Numerical methods such as implicit finite difference methods have been used to solve non-similar boundary layer equations by Hossain and Paul [156] while Hussain et al. [160] and Yih [42] used the Keller-box method. Finite difference methods require many grid points to achieve accurate results with consequent demands of extensive computational time. By contrast, pseudospectral methods converge to the exact solution using few grid points and hence are computationally faster, and have been used to solve boundary layer equations [161–164].

Some new pseudospectral methods have been recently developed to solve boundary layer equations. These include the spectral quasilinearization method, spectral local linearization method, bivariate spectral quasilinearization method, bivariate spectral local linearization method, and many more. Spectral quasilinearization methods (SQLM) has been used to solve the magnetohydrodynamic (MHD) boundary layer flow of an incompressible upper-convected Maxwell (UCM) fluid over a porous stretching surface by Motsa et al. [165]. Shateyi and Marewo [166] used the SQLM to solve the magnetohydrodynamic boundary layer flow equation with heat and mass transfer of an incompressible upper-convected Maxwell fluid over a stretching sheet in the presence of viscous dissipation and thermal radiation as well as chemical reaction. The method obtained second order accurate results within few seconds since it used few grid points. It converged very fast to the exact solution. However, the SQLM method solves a system of equations as coupled differential equations. Solving the system of matrices as a coupled system leads to a big matrix and hence, the more system of equations we solve, the bigger the matrix. Increasing the size of the matrix may lead to ill-conditioning of the matrix and hence inaccurate results.

An alternative method was developed by Motsa [167]. This method termed, the spectral local linearization method (SLLM), was developed based on, decoupling and linearizing systems of

equations using a combination of a univariate linearization technique, then spectral collocation discretization. The main feature of the SLLM algorithm is the breaking down of a large coupled system of equations into sequences of smaller subsystems which are then solved in a sequence form in a computationally efficient manner. The method has been applied to solve a coupled systems of equation that model the problem of unsteady free convective heat and mass transfer and the Blasius equation [167]. The results demonstrated that the method is accurate, convergent, stable, and very efficient when compared with other existing methods of solving some large systems of boundary value problems.

The SQLM method was extended for the first time to solve nonlinear partial differential equation by Motsa et al. [168]. Motsa et al. [168] solved a system of equations modeling boundary layer flow caused by an impulsively stretching plate and a coupled system of four nonlinear differential equations that model the problem of unsteady MHD flow and mass transfer in a porous space. Spectral methods were used in space and finite differences were used in time. Using finite differences in space increases the computational time and decreases the accuracy of the spectral method. In an attempt to correct the disadvantages due to the usage of finite differences in time, Motsa et al. [97] applied spectral method in both space and time to solve nonlinear evolution equations. The method was termed the bivariate spectral quasilinearization method (BSQLM). Abbas [169] solved systems of partial differential equations modeling the heat and mass transfer in an unsteady boundary layer flow of a Casson fluid near a stagnation point over a stretching/shrinking sheet in the presence of thermal radiation using the BSQLM. However, this method solves a system of equations as coupled system of equations. This in turn implies that the method results in a huge matrix which may suffer consequences of ill-conditioning. In an attempt to solve this problem, Motsa [170] developed the bivariate spectral local linearization method (BSLLM) to solve a flow model described in terms of a highly coupled and nonlinear system of partial differential equations that model the problem of unsteady mixed convection flow over a vertical cone due to impulsive motion. The BSLLM method has been extended to solve a system of differential equations modeling the behavior of unsteady non-Darcian magnetohydrodynamic fluid flow past an impulsively started vertical porous surface [171]. These methods converge fast and are computationally fast.

However, these bivariate pseudospectral methods are suitable for nonlinear partial differential equations with small time domain. Increasing the time domain reduces the accuracy of the

method. In some cases, the computed numerical solutions fail to converge even when the number of grid points are increased. Thus, accurate solutions of systems of differential equations are usually determined over short time intervals.

The realization that accurate solutions can be obtained in small time intervals have led to a number of numerical methods that uses the domain decomposition technique. The domain-decomposition approach together with the spectral collocation technique has been used to solve equations arising from various scientific fields. Funaro [172] and Harald [173] solved elliptic differential equations, Kopriva [174] solved hyberbolic systems, Peyret [175] solved stiff problems in fluid mechanics and recently, it has been used to solve chaotic system of ordinary differential equations by [176–179]. These methods split large intervals into smaller sub-intervals so that the equations are solved over a sequence of non-overlapping sub-intervals of the domain. Continuity conditions are used to advance the solution across the non-overlapping sub-intervals.

In this chapter, we present multi-domain numerical approaches for solutions of systems of nonlinear coupled non-similar boundary layer partial differential equations over a large time interval. The multi-domain bivariate spectral local linearisation method (MD-BSLLM), and multi-domain bivariate spectral quasilinearisation method (MD-BSQLM), are an extension of the BSLLM and BSQLM respectively. The proposed approaches are developed for solving system of n coupled non-similar boundary layer partial differential equations. The multi-domain approach are only applied in the time domain. The domain is divided into smaller non-overlapping sub-intervals on which the Chebyshev spectral collocation method is used to solve the equations. A continuity condition is used to advance the solution across the sub-intervals. The algorithms are easy to develop and yield accurate results using only a few discretization nodes. The accuracy of the methods is validated against the series solution for limiting cases.

The main aim of the study is to explore the applicability of the multi-domain bivariate spectral local linearisation method and the multi-domain bivariate spectral quasilinearisation method to systems of nonlinear coupled non-similar boundary layer partial differential equations over large time intervals. The results confirm that these methods are suitable for solving all types of systems of nonlinear coupled non-similar boundary layer partial differential equations over a large time interval.

The article is organized as follows, in Sections 8.2 and 8.3, the multi-domain bivariate spectral quasilinearisation method and the multi-domain bivariate spectral local linearisation method are described for a general system of n partial differential equations respectively. In Section 8.4, numerical experiments and the series solutions are presented in Section 8.5. In Section 8.6, the results are discussed and we conclude in Section 8.7.

8.2 Multi-Domain Bivariate Spectral Quasilinearisation Method

In this section we introduce the multi-domain bivariate spectral quasilinearization method for a general system of n nonlinear partial differential equations. We consider a system of the form,

$$\begin{aligned} \Phi_1[G_1, G_2, \ldots, G_n] &= 0, \\ \Phi_2[G_1, G_2, \ldots, G_n] &= 0, \\ &\vdots \\ \Phi_n[G_1, G_2, \ldots, G_n] &= 0, \end{aligned} \tag{8.1}$$

where the operators G_i for $i = 1, 2, \ldots, n$ are of the form

$$\begin{aligned} G_1 &= \left\{ f_1, \frac{\partial f_1}{\partial \eta}, \frac{\partial^2 f_1}{\partial \eta^2}, \ldots, \frac{\partial^p f_1}{\partial \eta^p}, \frac{\partial f_1}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_1}{\partial \eta}\right) \right\}, \\ G_2 &= \left\{ f_2, \frac{\partial f_2}{\partial \eta}, \frac{\partial^2 f_2}{\partial \eta^2}, \ldots, \frac{\partial^p f_2}{\partial \eta^p}, \frac{\partial f_2}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_2}{\partial \eta}\right) \right\}, \\ &\vdots \\ G_n &= \left\{ f_n, \frac{\partial f_n}{\partial \eta}, \frac{\partial^2 f_n}{\partial \eta^2}, \ldots, \frac{\partial^p f_n}{\partial \eta^p}, \frac{\partial f_n}{\partial \zeta}, \frac{\partial}{\partial \zeta}\left(\frac{\partial f_n}{\partial \eta}\right) \right\}. \end{aligned} \tag{8.2}$$

The order of differentiation is denoted by p, the required solution by $f_k(\eta, \zeta)$ and Φ_k for $k = 1, 2, \ldots, n$ are the non-linear operators containing all the spatial and time derivatives of $f_k(\eta, \zeta)$. The Legendre-Gauss-Lobbatto grid points and the corresponding differentiation matrices are defined in the interval $[-1, 1]$. Therefore, the time interval $\zeta \in [0, T]$ and space region $x \in [a, b]$, are transformed to $[-1, 1]$ using linear transformations. In order to apply the multi-domain bivariate spectral quasilinearization method, we first decompose the interval $\zeta \in [0, T]$ into q non-overlapping sub-intervals $\Omega_l = [\zeta_{l-1}, \zeta_l]$ for $l = 1, 2, \ldots, q$. Each interval

Ω_l is further divided into s not necessarily equal divisions. The system of non-linear partial differential equations are solved in each sub-interval $[\zeta_{l-1}, \zeta_l]$. The solutions in each sub-interval $[\zeta_{l-1}, \zeta_l]$ are denoted by $f_k^{(l)}(\eta, \zeta)$, for $k = 1, 2, \ldots, n$. In the first interval $[\zeta_0, \zeta_1]$, the solutions $f_k^{(l)}(\eta, \zeta)$, for $k = 1, 2, \ldots, n$ are obtained subject to the initial conditions $f_k^{(1)}(\eta, 0)$, for $k = 1, 2, \ldots, n$ respectively. For each $l > 1$, at each sub-interval $[\zeta_{l-1}, \zeta_l]$, the continuity conditions

$$f_k^{(l)}(\eta, \zeta_{l-1}) = f_k^{(l-1)}(\eta, \zeta_{l-1}), \quad \text{for} \quad k = 1, 2, \ldots, n \quad \text{and} \quad l = 1, 2, \ldots, q, \tag{8.3}$$

are used to implement the MD-BSQLM over the interval $[\zeta_{l-1}, \zeta_l]$. This procedure is repeated to generate a sequence of solutions $f_k^{(l)}(\eta, \zeta)$, for $l = 1, 2, \ldots, p$ and $k = 1, 2, \ldots, n$. We assume that at each sub-interval $[\zeta_{l-1}, \zeta_l]$, the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$f_k^{(l)}(\eta, \zeta) \approx \sum_{i=0}^{N_\eta} \sum_{j=0}^{N_\zeta} f_k^{(l)}(\eta_i, \zeta_j) \mathscr{L}_i(\eta) \mathscr{L}_j(\zeta), \tag{8.4}$$

for $k = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, p$. The bivariate Lagrange interpolation polynomial interpolates $f_k^{(l)}(\eta, \zeta)$ at selected points called Legendre-Gauss-Lobatto points (η_i, ζ_j) in both the η and ζ directions, for $i = 0, 1, 2, \ldots, N_\eta$ and $j = 0, 1, 2, \ldots, N_\zeta$. The nonlinear operators Φ_k, for $k = 1, 2, 3, \ldots, n$ are first linearized using the quasilinearisation technique as defined by Bellman and Kalaba [71]. The quasilinearisation method is a technique based on the Taylor series expansion of Φ_k about some previous iteration. We assume that the difference between it's previous and current solution and all it's derivatives are small. Applying the quasilinearisation method yields the following

$$\begin{aligned} \Phi_k[G_1, G_2, \ldots, G_n] \approx (G_{1,r+1} - G_{1,r}, G_{2,r+1} - G_{2,r}, \ldots, G_{n,r+1} - G_{n,r}) \cdot \nabla \Phi_k[G_{1,r}, G_{2,r}, \ldots, G_{n,r}] \\ + \Phi_k[G_{1,r}, G_{2,r}, \ldots, G_{n,r}], \end{aligned} \tag{8.5}$$

where r and $r+1$ denote previous and current iterations respectively and ∇ is a vector of the partial derivatives which is defined as

$$\nabla = \left\{ \nabla_{f_1}, \nabla_{f_2}, \ldots, \nabla_{f_n} \right\}. \tag{8.6}$$

We define

$$\begin{aligned}
\nabla_{f_1} &= \left\{ \frac{\partial}{\partial f_1}, \frac{\partial}{\partial f_1'}, \frac{\partial}{\partial f_1''}, \ldots, \frac{\partial}{\partial f_1^{(p)}}, \frac{\partial}{\partial \left(\frac{\partial f_1}{\partial \zeta}\right)}, \frac{\partial}{\partial \left(\frac{\partial f_1'}{\partial \zeta}\right)} \right\}, \\
\nabla_{f_2} &= \left\{ \frac{\partial}{\partial f_2}, \frac{\partial}{\partial f_2'}, \frac{\partial}{\partial f_2''}, \ldots, \frac{\partial}{\partial f_2^{(p)}}, \frac{\partial}{\partial \left(\frac{\partial f_2}{\partial \zeta}\right)}, \frac{\partial}{\partial \left(\frac{\partial f_2'}{\partial \zeta}\right)} \right\}, \\
&\vdots \\
\nabla_{f_n} &= \left\{ \frac{\partial}{\partial f_n}, \frac{\partial}{\partial f_n'}, \frac{\partial}{\partial f_n''}, \ldots, \frac{\partial}{\partial f_n^{(p)}}, \frac{\partial}{\partial \left(\frac{\partial f_n}{\partial \zeta}\right)}, \frac{\partial}{\partial \left(\frac{\partial f_n'}{\partial \zeta}\right)} \right\},
\end{aligned} \tag{8.7}$$

where the prime denotes differentiation with respect to η. The linearised equation (8.5) can be expressed in a compact form as

$$\sum_{s=1}^{n} G_{s,r+1} \cdot \nabla_{f_s} \Phi_k \left[G_{1,r}, G_{2,r}, \ldots, G_{n,r} \right] = \sum_{s=1}^{n} G_{s,r} \cdot \nabla_{f_s} \Phi_k \left[G_{1,r}, G_{2,r}, \ldots, G_{n,r} \right] - \Phi_k \left[G_{1,r}, G_{2,r}, \ldots, G_{n,r} \right] \tag{8.8}$$

for $k = 1, 2, \ldots, n$. Equation (8.8) forms a system of n coupled linear partial differential equations. They are solved iteratively for $f_{1,r+1}^{(l)}(\eta,\zeta), f_{2,r+1}^{(l)}(\eta,\zeta), \ldots, f_{n,r+1}^{(l)}(\eta,\zeta)$. Equation (8.8) can further be expressed as:

$$\begin{aligned}
\sum_{v=1}^{n} \left[\sum_{s=0}^{p} \alpha_{v,s,r}^{(l,1)}(\eta,\zeta) f_{v,r+1}^{(s,l)} + \beta_{v,r}^{(l,1)}(\eta,\zeta) \frac{\partial f_{v,r+1}^{(l)}}{\partial \zeta} + \gamma_{v,r}^{(l,1)}(\eta,\zeta) \frac{\partial}{\partial \zeta} \left(\frac{\partial f_{v,r+1}^{(l)}}{\partial \eta} \right) \right] &= R_1^{(l)}(\eta,\zeta), \\
\sum_{v=1}^{n} \left[\sum_{s=0}^{p} \alpha_{v,s,r}^{(l,2)}(\eta,\zeta) f_{v,r+1}^{(s,l)} + \beta_{v,r}^{(l,2)}(\eta,\zeta) \frac{\partial f_{v,r+1}^{(l)}}{\partial \zeta} + \gamma_{v,r}^{(l,2)}(\eta,\zeta) \frac{\partial}{\partial \zeta} \left(\frac{\partial f_{v,r+1}^{(l)}}{\partial \eta} \right) \right] &= R_2^{(l)}(\eta,\zeta), \\
\sum_{v=1}^{n} \left[\sum_{s=0}^{p} \alpha_{v,s,r}^{(l,3)}(\eta,\zeta) f_{v,r+1}^{(s,l)} + \beta_{v,r}^{(l,3)}(\eta,\zeta) \frac{\partial f_{v,r+1}^{(l)}}{\partial \zeta} + \gamma_{v,r}^{(l,3)}(\eta,\zeta) \frac{\partial}{\partial \zeta} \left(\frac{\partial f_{v,r+1}^{(l)}}{\partial \eta} \right) \right] &= R_3^{(l)}(\eta,\zeta), \\
&\vdots \\
\sum_{v=1}^{n} \left[\sum_{s=0}^{p} \alpha_{v,s,r}^{(l,n)}(\eta,\zeta) f_{v,r+1}^{(s,l)} + \beta_{v,r}^{(l,n)}(\eta,\zeta) \frac{\partial f_{v,r+1}^{(l)}}{\partial \zeta} + \gamma_{v,r}^{(l,n)}(\eta,\zeta) \frac{\partial}{\partial \zeta} \left(\frac{\partial f_{v,r+1}^{(l)}}{\partial \eta} \right) \right] &= R_n^{(l)}(\eta,\zeta),
\end{aligned} \tag{8.9}$$

where $\alpha_{n,p,r}^{(l,k)}(\eta,\zeta)$, $\beta_{v,r}^{(l,k)}(\eta,\zeta)$ and $\gamma_{v,r}^{(l,k)}(\eta,\zeta)$ are variable coefficients of $f_{n,r+1}^{(p,l)}$, $\frac{\partial f_{v,r+1}^{(l)}}{\partial \zeta}$, and $\frac{\partial}{\partial \zeta}\left(\frac{\partial f_{v,r+1}^{(l)}}{\partial \eta}\right)$, respectively in the l^{th} sub-interval of the multi-domain approach. These variable coefficients correspond to the k^{th} equation, for $k = 1, 2, \ldots, n$. The constant p denotes the order

of differentiation. Thus, we have

$$\alpha_{n,p,r}^{(l,k)}(\eta,\zeta)=\frac{\partial\Phi_k}{\partial f_{n,r}^{(p,l)}},\quad \beta_{v,r}^{(l,k)}(\eta,\zeta)=\frac{\partial\Phi_k}{\partial\left(\frac{\partial f_{v,r}^{(l)}}{\partial\zeta}\right)},\quad \gamma_{v,r}^{(l,k)}(\eta,\zeta)=\frac{\partial\Phi_k}{\partial\left(\frac{\partial}{\partial\zeta}\left(\frac{\partial f_{v,r}^{(l)}}{\partial\eta}\right)\right)}. \tag{8.10}$$

Collocating equation (8.9) yields

$$\begin{aligned}
&\sum_{v=1}^{n}\left[A_{1,v}^{(i,l)}\mathbf{F}_{v,i}^{(l)}+\boldsymbol{\beta}_{v,r}^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}^{(l)}+\boldsymbol{\gamma}_{v,r}^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}^{(l)}\right]=\mathbf{R}_{1,i}^{(l)},\\
&\sum_{v=1}^{n}\left[A_{2,v}^{(i,l)}\mathbf{F}_{v,i}^{(l)}+\boldsymbol{\beta}_{v,r}^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}^{(l)}+\boldsymbol{\gamma}_{v,r}^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}^{(l)}\right]=\mathbf{R}_{2,i}^{(l)},\\
&\sum_{v=1}^{n}\left[A_{3,v}^{(i,l)}\mathbf{F}_{v,i}^{(l)}+\boldsymbol{\beta}_{v,r}^{(l,3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}^{(l)}+\boldsymbol{\gamma}_{v,r}^{(l,3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}^{(l)}\right]=\mathbf{R}_{3,i}^{(l)},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&\sum_{v=1}^{n}\left[A_{n,v}^{(i,l)}\mathbf{F}_{v,i}^{(l)}+\boldsymbol{\beta}_{v,r}^{(l,n)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{v,j}^{(l)}+\boldsymbol{\gamma}_{v,r}^{(l,n)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{v,j}^{(l)}\right]=\mathbf{R}_{n,i}^{(l)},
\end{aligned} \tag{8.11}$$

where

$$A_{k,v}^{(i,l)}=\sum_{s=0}^{p}\boldsymbol{\alpha}_{v,s,r}^{(l,k)}\mathbf{D}^{(s)}, \tag{8.12}$$

for $k, v = 1, 2, \ldots, n$, and

$$\boldsymbol{\alpha}_{v,s,r}^{(l,k)} = \begin{bmatrix} \alpha_{v,s,r}^{(l,k)}(\eta_0, \zeta_j) & & & \\ & \alpha_{v,s,r}^{(l,k)}(\eta_1, \zeta_j) & & \\ & & \ddots & \\ & & & \alpha_{v,s,r}^{(l,k)}(\eta_{N_\eta}, \zeta_j) \end{bmatrix}, \tag{8.13}$$

$$\boldsymbol{\beta}_{v,r}^{(l,k)} = \begin{bmatrix} \beta_{v,r}^{(l,k)}(\eta_0, \zeta_j) & & & \\ & \beta_{v,r}^{(l,k)}(\eta_1, \zeta_j) & & \\ & & \ddots & \\ & & & \beta_{v,r}^{(l,k)}(\eta_{N_\eta}, \zeta_j) \end{bmatrix}, \tag{8.14}$$

$$\boldsymbol{\gamma}_{v,r}^{(l,k)} = \begin{bmatrix} \gamma_{v,r}^{(l,k)}(\eta_0, \zeta_j) & & & \\ & \gamma_{v,r}^{(l,k)}(\eta_1, \zeta_j) & & \\ & & \ddots & \\ & & & \gamma_{v,r}^{(l,k)}(\eta_{N_\eta}, \zeta_j) \end{bmatrix}. \tag{8.15}$$

Equation (8.11) is expressed in the following matrix form

$$\mathbf{B}_r \Gamma_{r+1} = \mathbf{R}_r \tag{8.16}$$

where the coefficient matrix $\mathbf{B}_r$ is defined as

$$\left[\begin{array}{cccc|cccc|ccc|cccc}
B_{1,1}^{(0,0)} & B_{1,2}^{(0,0)} & \cdots & B_{1,n}^{(0,0)} & B_{1,1}^{(0,1)} & B_{1,2}^{(0,1)} & \cdots & B_{1,n}^{(0,1)} & \ddots & & & B_{1,1}^{(0,N_\eta)} & B_{1,2}^{(0,N_\eta)} & \cdots & B_{1,n}^{(0,N_\eta)} \\
B_{2,1}^{(0,0)} & B_{2,2}^{(0,0)} & \cdots & B_{2,n}^{(0,0)} & B_{2,1}^{(0,1)} & B_{2,2}^{(0,1)} & \cdots & B_{2,n}^{(0,1)} & & \ddots & & B_{2,1}^{(0,N_\eta)} & B_{2,2}^{(0,N_\eta)} & \cdots & B_{2,n}^{(0,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(0,0)} & B_{n,2}^{(0,0)} & \cdots & B_{n,n}^{(0,0)} & B_{n,1}^{(0,1)} & B_{n,2}^{(0,1)} & \cdots & B_{n,n}^{(0,1)} & & & \ddots & B_{n,1}^{(0,N_\eta)} & B_{n,2}^{(0,N_\eta)} & \cdots & B_{n,n}^{(0,N_\eta)} \\
\hline
B_{1,1}^{(1,0)} & B_{1,2}^{(1,0)} & \cdots & B_{1,n}^{(1,0)} & B_{1,1}^{(1,1)} & B_{1,2}^{(1,1)} & \cdots & B_{1,n}^{(1,1)} & \ddots & & & B_{1,1}^{(1,N_\eta)} & B_{1,2}^{(1,N_\eta)} & \cdots & B_{1,n}^{(1,N_\eta)} \\
B_{2,1}^{(1,0)} & B_{2,2}^{(1,0)} & \cdots & B_{2,n}^{(1,0)} & B_{2,1}^{(1,1)} & B_{2,2}^{(1,1)} & \cdots & B_{2,n}^{(1,1)} & & \ddots & & B_{2,1}^{(1,N_\eta)} & B_{2,2}^{(1,N_\eta)} & \cdots & B_{2,n}^{(1,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(1,0)} & B_{n,2}^{(1,0)} & \cdots & B_{n,n}^{(1,0)} & B_{n,1}^{(1,1)} & B_{n,2}^{(1,1)} & \cdots & B_{n,n}^{(1,1)} & & & \ddots & B_{n,1}^{(1,N_\eta)} & B_{n,2}^{(1,N_\eta)} & \cdots & B_{n,n}^{(1,N_\eta)} \\
\hline
\ddots & & & & \ddots & & & & \cdots & \cdots & \cdots & \ddots & & & \\
& \ddots & & & & \ddots & & & \cdots & \cdots & \cdots & & \ddots & & \\
& & \ddots & & & & \ddots & & \cdots & \cdots & \cdots & & & \ddots & \\
& & & \ddots & & & & \ddots & \cdots & \cdots & \cdots & & & & \ddots \\
\hline
B_{1,1}^{(N_\eta,0)} & B_{1,2}^{(N_\eta,0)} & \cdots & B_{1,n}^{(N_\eta,0)} & B_{1,1}^{(N_\eta,1)} & B_{1,2}^{(N_\eta,1)} & \cdots & B_{1,n}^{(N_\eta,1)} & \ddots & & & B_{1,1}^{(N_\eta,N_\eta)} & B_{1,2}^{(N_\eta,N_\eta)} & \cdots & B_{1,n}^{(N_\eta,N_\eta)} \\
B_{2,1}^{(N_\eta,0)} & B_{2,2}^{(N_\eta,0)} & \cdots & B_{2,n}^{(N_\eta,0)} & B_{2,1}^{(N_\eta,1)} & B_{2,2}^{(N_\eta,1)} & \cdots & B_{2,n}^{(N_\eta,1)} & & \ddots & & B_{2,1}^{(N_\eta,N_\eta)} & B_{2,2}^{(N_\eta,N_\eta)} & \cdots & B_{2,n}^{(N_\eta,N_\eta)} \\
\vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots & & & \ddots & \vdots & \vdots & \cdots & \vdots \\
B_{n,1}^{(N_\eta,0)} & B_{n,2}^{(N_\eta,0)} & \cdots & B_{n,n}^{(N_\eta,0)} & B_{n,1}^{(N_\eta,1)} & B_{n,2}^{(N_\eta,1)} & \cdots & B_{n,n}^{(N_\eta,1)} & & & \ddots & B_{n,1}^{(N_\eta,N_\eta)} & B_{n,2}^{(N_\eta,N_\eta)} & \cdots & B_{n,n}^{(N_\eta,N_\eta)}
\end{array}\right]$$

and the entries are defined as

$$\begin{aligned}
B_{k,\ell}^{(i,i)} &= A_{v,k}^{i,l} + \boldsymbol{\beta}_{v,r}^{(1,k)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_{v,k}^{(1,k)} d_{i,i}\mathbf{D}, \quad \text{for } v,k = 1,2,\ldots,n, \ \ l = 1,2,\ldots,p, \ \text{ when } i = j, \\
B_{k,\ell}^{(i,j)} &= \boldsymbol{\beta}_{v,r}^{(1,k)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_{v,k}^{(1,k)} d_{i,j}\mathbf{D}, \quad \text{for } v,k = 1,2,\ldots,n, \ \ l = 1,2,\ldots,p, \ \text{ when } i \neq j
\end{aligned} \tag{8.17}$$

The vectors Γ_{r+1} and $\mathbf{R}_r$ are defined as

$$\Gamma_{r+1} = \left[\mathbf{F}_{1,r+1}^{(0)}\mathbf{F}_{2,r+1}^{(0)}\cdots\mathbf{F}_{n,r+1}^{(0)} \,\middle|\, \mathbf{F}_{1,r+1}^{(1)}\mathbf{F}_{2,r+1}^{(1)}\cdots\mathbf{F}_{n,r+1}^{(1)} \,\middle|\, \cdots\cdots\cdots\cdots \,\middle|\, \mathbf{F}_{1,r+1}^{(N_\zeta)}\mathbf{F}_{2,r+1}^{(N_\zeta)}\cdots\mathbf{F}_{n,r+1}^{(N_\zeta)}\right]^T \tag{8.18}$$

$$\mathbf{R}_r = \left[\mathbf{R}_{1,r}^{(0)}\ \mathbf{R}_{2,r}^{(0)}\ \ \mathbf{R}_{3,r}^{(0)}\cdots\mathbf{R}_{n,r}^{(0)} \,\middle|\, \mathbf{R}_{1,r}^{(1)}\ \ \mathbf{R}_{2,r}^{(1)}\ \ \mathbf{R}_{3,r}^{(1)}\cdots\mathbf{R}_{n,r}^{(1)} \,\middle|\, \cdots\cdots\cdots\cdots \,\middle|\, \mathbf{R}_{1,r}^{(N_\zeta)}\ \mathbf{R}_{2,r}^{(N_\zeta)}\ \cdots\ \mathbf{R}_{n,r}^{(N_\zeta)}\right]^T \tag{8.19}$$

Multiplying equation (8.16) by the inverse of the matrix $\mathbf{B}_r$ yields the solution to equation (8.1).

8.3 Multi-domain Bivariate Spectral Local Linearisation Method

In this section we introduce the multi-domain bivariate spectral local linearisation method (MD-BSLLM) for approximating solutions of system of nonlinear partial differential equations

that model non-similar boundary layer equations. Without loss of generality, we consider system of n nonlinear partial differential equations of the form (8.1). Applying quasilinearisation independently on each equation, we get

$$\begin{aligned}
&\sum_{s=0}^{p} \alpha_{s,r}^{(l,1)}(\eta,\zeta) f_{1,r+1}^{(s,l)} + \beta_r^{(l,1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(0,l)}}{\partial \zeta} + \gamma_r^{(l,1)}(\eta,\zeta)\frac{\partial f_{1,r+1}^{(1,l)}}{\partial \zeta} = R_1^{(l)}(\eta,\zeta),\\
&\sum_{s=0}^{p} \alpha_{s,r}^{(l,2)}(\eta,\zeta) f_{2,r+1}^{(s,l)} + \beta_r^{(l,2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(0,l)}}{\partial \zeta} + \gamma_r^{(l,2)}(\eta,\zeta)\frac{\partial f_{2,r+1}^{(1,l)}}{\partial \zeta} = R_2^{(l)}(\eta,\zeta),\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots\\
&\sum_{s=0}^{p} \alpha_{s,r}^{(l,n)}(\eta,\zeta) f_{n,r+1}^{(s,l)} + \beta_r^{(l,n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(0,l)}}{\partial \zeta} + \gamma_r^{(l,n)}(\eta,\zeta)\frac{\partial f_{n,r+1}^{(1,l)}}{\partial \zeta} = R_n^{(l)}(\eta,\zeta),
\end{aligned} \tag{8.20}$$

where $\alpha_{s,r}^{(l,k)}(\eta,\zeta)$, $\beta_r^{(l,k)}(\eta,\zeta)$ and $\gamma_r^{(l,k)}(\eta,\zeta)$ are variable coefficients of $f_{k,r+1}^{(s,l)}$, $\frac{\partial f_{k,r+1}^{(0,l)}}{\partial \zeta}$, and $\frac{\partial f_{k,r+1}^{(1,l)}}{\partial \zeta}$, respectively in the l^{th} sub-interval of the multi-domain approach, for $k = 1,2,\ldots,n$ and $s = 0,1,2,\ldots,p$. These variable coefficients correspond to the k^{th} equation, for $k = 1,2,\ldots,n$. The constant p denotes the order of differentiation. Thus, we have

$$\alpha_{s,r}^{(l,k)}(\eta,\zeta) = \frac{\partial \Phi_k}{\partial f_{k,r}^{(s,l)}}, \qquad \beta_r^{(l,k)}(\eta,\zeta) = \frac{\partial \Phi_k}{\partial\left(\frac{\partial f_{k,r}^{(0,l)}}{\partial \zeta}\right)}, \qquad \gamma_r^{(l,k)}(\eta,\zeta) = \frac{\partial \Phi_k}{\partial\left(\frac{\partial f_{k,r}^{(1,l)}}{\partial \zeta}\right)}. \tag{8.21}$$

In general, the kth right hand side is given by

$$R_k^{(l)}(\eta,\zeta) = \sum_{s=0}^{p} \alpha_{s,r}^{(l,k)}(\eta,\zeta) f_{k,r}^{(s,l)} + \beta_r^{(l,k)}(\eta,\zeta)\frac{\partial f_{k,r}^{(0,l)}}{\partial \zeta} + \gamma_r^{(l,k)}(\eta,\zeta)\frac{\partial f_{k,r}^{(1,l)}}{\partial \zeta)} - \Phi_k. \tag{8.22}$$

Applying collocation on equation (8.20), we obtain

$$\begin{aligned}
&A_{1,1}^{(l)}\mathbf{F}_{1,i}^{(l)}+\boldsymbol{\beta}_r^{(l,1)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{1,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,1)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{1,j}^{(l)}=\mathbf{R}_{1,i}^{(l)},\\
&A_{2,2}^{(l)}\mathbf{F}_{2,i}^{(l)}+\boldsymbol{\beta}_r^{(l,2)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{2,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,2)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{2,j}^{(l)}=\mathbf{R}_{2,i}^{(l)},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\vdots\\
&A_{n,n}^{(l)}\mathbf{F}_{n,i}^{(l)}+\boldsymbol{\beta}_r^{(l,n)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{F}_{n,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,n)}\sum_{j=0}^{N_\zeta} d_{i,j}\mathbf{DF}_{n,j}^{(l)}=\mathbf{R}_{n,i}^{(l)},
\end{aligned} \tag{8.23}$$

where

$$A_{1,1}^{(l)}=\sum_{s=0}^{p}\boldsymbol{\alpha}_{s,r}^{(l,1)}\mathbf{D}^{(s)},\quad A_{2,2}^{(l)}=\sum_{s=0}^{p}\boldsymbol{\alpha}_{s,r}^{(l,2)}\mathbf{D}^{(s)},\quad \ldots,\quad A_{n,n}^{(l)}=\sum_{s=0}^{p}\boldsymbol{\alpha}_{s,r}^{(l,n)}\mathbf{D}^{(s)}. \tag{8.24}$$

The diagonal matrices of the corresponding variable coefficients are given by

$$\boldsymbol{\alpha}_{s,r}^{(l,k)}=\begin{bmatrix}\alpha_{s,r}^{(l,k)}(\eta_0,\zeta_j) & & & \\ & \alpha_{s,r}^{(l,k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \alpha_{s,r}^{(l,k)}(\eta_{N_\eta},\zeta_j)\end{bmatrix}, \tag{8.25}$$

$$\boldsymbol{\beta}_r^{(l,k)}=\begin{bmatrix}\beta_r^{(l,k)}(\eta_0,\zeta_j) & & & \\ & \beta_r^{(l,k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \beta_r^{(l,k)}(\eta_{N_\eta},\zeta_j)\end{bmatrix}, \tag{8.26}$$

$$\boldsymbol{\gamma}_r^{(l,k)}=\begin{bmatrix}\gamma_r^{(l,k)}(\eta_0,\zeta_j) & & & \\ & \gamma_r^{(l,k)}(\eta_1,\zeta_j) & & \\ & & \ddots & \\ & & & \gamma_r^{(l,k)}(\eta_{N_\eta},\zeta_j)\end{bmatrix}. \tag{8.27}$$

Imposing boundary conditions for $i=0,1,\cdots,N_\zeta-1$, each equation in (8.23) can be expressed as the following $N_\zeta(N_\eta+1)\times N_\zeta(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0}^{(l,k)} & B_{0,1}^{(l,k)} & \cdots & B_{0,N_\zeta-1}^{(l,k)} \\ B_{1,0}^{(l,k)} & B_{1,1}^{(l,k)} & \cdots & B_{1,N_\zeta-1}^{(l,k)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,k)} & B_{N_\zeta-1,1}^{(l,k)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,k)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{k,0}^{(l)} \\ \mathbf{F}_{k,1}^{(l)} \\ \vdots \\ \mathbf{F}_{k,N_\zeta-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{k,0}^{(l)} \\ \mathscr{R}_{k,1}^{(l)} \\ \vdots \\ \mathscr{R}_{k,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.28}$$

where

$$\begin{aligned} B_{(i,i)}^{(l,k)} &= \sum_{s=0}^{p} \boldsymbol{\alpha}_{s,r}^{(l,k)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,k)} d_{i,i} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,k)} d_{i,i} \mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \ l = 1,2,\ldots,z, \ \text{when } i = j, \\ B_{(i,j)}^{(l,k)} &= \boldsymbol{\beta}_r^{(l,k)} d_{i,j} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,k)} d_{i,j} \mathbf{D}, \quad \text{for } k = 1,2,\ldots,n, \ l = 1,2,\ldots,z, \ \text{when } i \neq j. \end{aligned} \tag{8.29}$$

The vector $\mathscr{R}_{k,i}^{(l)}$ is defined as

$$\mathscr{R}_{k,i}^{(l)} = \mathbf{R}_{k,i}^{(l)} - \left(\boldsymbol{\beta}_r^{(l,k)} d_{i,N_\zeta} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,k)} d_{i,N_\zeta} \mathbf{D} \right) \mathbf{F}_{k,N_\zeta}^{(l)}$$

for $i = 0,1,\cdots,N_\zeta - 1$ and $k = 1,2,\ldots,n$. The vector $\mathbf{F}_{k,N_\zeta}^{(l)}$ corresponds to the initial boundary condition which is always prescribed. Equation (8.28), is of the form

$$\mathbf{B}^{(l,k)} \mathbf{F}_k^{(l)} = \mathbf{R}_k^{(l)} \tag{8.30}$$

for $k = 1,2,3,\ldots,n$. Multiplying equation (8.30) by the inverse of $\mathbf{B}^{(l,k)}$ yields the solution of equations (8.1).

8.4 Numerical Experiments

In this section, we present systems of equation for various fluid flow models considered in this study. The MD-BSLLM and MD-BSQLM algorithms are applied to each system of equations as a numerical experiment.

Steady free convection flow past a non-isothermal vertical porous cone

In this subsection, we consider the problem of steady two-dimensional laminar free convection flow past a non-isothermal vertical porous cone with variable surface temperature is also

considered for numerical experimentation. The governing non-similarity system of partial differential equations are expressed in dimensionless form as (see Hossain et. al. [156] for derivation)

$$f'''+\frac{n+7}{4}ff''-\frac{n+1}{2}f'^2+\theta+\zeta f''=\frac{1-n}{4}\zeta\left(f'\frac{\partial f'}{\partial\zeta}-f''\frac{\partial f}{\partial\zeta}\right), \tag{8.31}$$

$$\frac{1}{Pr}\theta''+\frac{n+7}{4}f\theta'-nf'\theta+\zeta\theta'=\frac{1-n}{4}\zeta\left(f'\frac{\partial\theta}{\partial\zeta}-\theta'\frac{\partial f}{\partial\zeta}\right), \tag{8.32}$$

where $Pr=\nu/\alpha$ is the Prandtl number, ζ is the dimensionless suction parameter, $\theta(\eta,\zeta)$ and $f(\eta,\zeta)$ are the dimensionless temperature and stream functions respectively. The appropriate corresponding boundary conditions are;

$$f(\zeta,0)=0,\quad f'(\zeta,0)=0,\quad \theta(\zeta,0)=1,\quad f'(\zeta,\infty)=0,\quad \theta(\zeta,\infty)=0. \tag{8.33}$$

The skin friction coefficient C_{fx} and the Nusselt number Nu_x describe the shear-stress and heat flux rate at the surface, respectively, and are defined by [156] as:

$$C_{fx}Gr_x^{1/4}=f''(\zeta,0),\quad \frac{Nu_x}{Gr_x^{1/4}}=-\theta'(\zeta,0). \tag{8.34}$$

In this case, the order of differentiation $p=3$ and the number of system of equations $n=2$. For the MD-BSLLM, we have

$$\begin{aligned}
A_{1,1}^{(l)}\mathbf{F}_{1,i}^{(l)}+\boldsymbol{\beta}_r^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{1,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{1,j}^{(l)}&=\mathbf{R}_{1,i}^{l},\\
A_{2,2}^{(l)}\mathbf{F}_{2,i}^{(l)}+\boldsymbol{\beta}_r^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{2,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{2,j}^{(l)}&=\mathbf{R}_{2,i}^{l},
\end{aligned} \tag{8.35}$$

where

$$A_{1,1}^{(l)}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(l,1)}\mathbf{D}^{(s)}=\boldsymbol{\alpha}_{3,r}^{(l,1)}\mathbf{D}^{(3)}+\boldsymbol{\alpha}_{2,r}^{(l,1)}\mathbf{D}^{(2)}+\boldsymbol{\alpha}_{1,r}^{(l,1)}\mathbf{D}^{(1)}+\boldsymbol{\alpha}_{0,r}^{(l,1)}, \tag{8.36}$$

$$A_{2,2}^{(l)}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(l,2)}\mathbf{D}^{(s)}=\boldsymbol{\alpha}_{3,r}^{(l,2)}\mathbf{D}^{(3)}+\boldsymbol{\alpha}_{2,r}^{(l,2)}\mathbf{D}^{(2)}+\boldsymbol{\alpha}_{1,r}^{(l,2)}\mathbf{D}^{(1)}+\boldsymbol{\alpha}_{0,r}^{(l,2)}. \tag{8.37}$$

If we let $f(\eta,\zeta)=f_1(\eta,\zeta)$ and $\theta(\eta,\zeta)=f_2(\eta,\zeta)$, then equations (8.31) and (8.32) can be expressed as

$$\Phi_1 = f_{1,r}^{(3,l)} + \frac{n+7}{4} f_{1,r}^{(0,l)} f_{1,r}^{(2,l)} - \frac{n+1}{2}\left(f_{1,r}^{(1,l)}\right)^2 + f_{2,r}^{(0,l)} + \zeta f_{1,r}^{(2,l)} \tag{8.38}$$
$$-\frac{1-n}{4}\zeta\left(f_{1,r}^{(1,l)}\frac{\partial f_{1,r}^{(1,l)}}{\partial\zeta} - f_{1,r}^{(2,l)}\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right),$$

$$\Phi_2 = \frac{1}{Pr} f_{2,r}^{(2,l)} + \frac{n+7}{4} f_{1,r}^{(0,l)} f_{2,r}^{(1,l)} - n f_{1,r}^{(1,l)} f_{2,r}^{(0,l)} + \zeta f_{2,r}^{(1,l)} - \frac{1-n}{4}\zeta\left(f_{1,r}^{(1,l)}\frac{\partial f_{2,r}^{(0,l)}}{\partial\zeta} - f_{2,r}^{(1,l)}\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right). \tag{8.39}$$

The variable coefficients in (8.36) are given by,

$$\alpha_{3,r}^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial f_{1,r}^{(3,l)}} = 1, \qquad \alpha_{2,r}^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial f_{1,r}^{(2,l)}} = \frac{1}{4}(n+7) f_{1,r}^{(1,l)} + \zeta,$$
$$\alpha_{1,r}^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial f_{1,r}^{(1,l)}} = -(n+1) f_{1,r}^{(1,l)}, \qquad \alpha_{0,r}^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial f_{1,r}^{(0,l)}} = \frac{1}{4}(n+7) f_{1,r}^{(2,l)},$$
$$\beta_r^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial\left(\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right)} = \frac{1}{4}(1-n)\zeta f_{1,r}^{(2,l)}, \qquad \gamma_r^{(l,1)}(\eta,\zeta) = \frac{\partial\Phi_1}{\partial\left(\frac{\partial f_{1,r}^{(1,l)}}{\partial\zeta}\right)} = -\frac{1}{4}(1-n)\zeta f_{1,r}^{(1,l)}.$$

and the variable coefficients for (8.37) are given by,

$$\gamma_r^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial\left(\frac{\partial f_{2,r}^{(1,l)}}{\partial\zeta}\right)} = 0, \qquad \alpha_{3,r}^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial f_{2,r}^{(3,l)}} = 0, \qquad \alpha_{2,r}^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial f_{2,r}^{(2,l)}} = \frac{1}{Pr},$$
$$\alpha_{1,r}^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial f_{2,r}^{(1,l)}} = \frac{1}{4}(n+7) f_{1,r}^{(0,l)} + \zeta + \frac{1}{4}(1-n)\zeta\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}, \qquad \alpha_{0,r}^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial f_{2,r}^{(0,l)}} = -n f_{1,r}^{(1,l)},$$
$$\beta_r^{(l,2)}(\eta,\zeta) = \frac{\partial\Phi_2}{\partial\left(\frac{\partial f_{2,r}^{(0,l)}}{\partial\zeta}\right)} = -\frac{1}{4}(1-n)\zeta f_{1,r}^{(1,l)}.$$

Equation (8.35) can be expressed as the following $N_\zeta(N_\eta+1)\times N_\zeta(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0}^{(l,1)} & B_{0,1}^{(l,1)} & \cdots & B_{0,N_\zeta-1}^{(l,1)} \\ B_{1,0}^{(l,1)} & B_{1,1}^{(l,1)} & \cdots & B_{1,N_\zeta-1}^{(l,1)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,1)} & B_{N_\zeta-1,1}^{(l,1)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,1)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{1,0}^{(l)} \\ \mathbf{F}_{1,1}^{(l)} \\ \vdots \\ \mathbf{F}_{1,N_\zeta-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{1,0}^{(l)} \\ \mathscr{R}_{1,1}^{(l)} \\ \vdots \\ \mathscr{R}_{1,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.40}$$

$$\begin{bmatrix} B_{0,0}^{(l,2)} & B_{0,1}^{(l,2)} & \cdots & B_{0,N_\zeta-1}^{(l,2)} \\ B_{1,0}^{(l,2)} & B_{1,1}^{(l,2)} & \cdots & B_{1,N_\zeta-1}^{(l,2)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,2)} & B_{N_\zeta-1,1}^{(l,2)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,2)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{2,0}^{(l)} \\ \mathbf{F}_{2,1}^{(l)} \\ \vdots \\ \mathbf{F}_{2,N_\zeta-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{2,0}^{(l)} \\ \mathscr{R}_{2,1}^{(l)} \\ \vdots \\ \mathscr{R}_{2,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.41}$$

where

$$\begin{aligned} B_{(i,i)}^{(l,1)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(l,1)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,1)} d_{i,i} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,i} \mathbf{D}, \quad \text{for } l = 1,2,\ldots,z \text{ when } i = j, \\ B_{(i,j)}^{(l,1)} &= \boldsymbol{\beta}_r^{(l,1)} d_{i,j} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,j} \mathbf{D}, \quad \text{for, } l = 1,2,\ldots,z \text{ when } i \neq j, \end{aligned} \tag{8.42}$$

$$\begin{aligned} B_{(i,i)}^{(l,2)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(l,2)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,2)} d_{i,i} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,i} \mathbf{D}, \quad \text{for } l = 1,2,\ldots,z \text{ when } i = j, \\ B_{(i,j)}^{(l,2)} &= \boldsymbol{\beta}_r^{(l,2)} d_{i,j} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,j} \mathbf{D}, \quad \text{for, } l = 1,2,\ldots,z \text{ when } i \neq j. \end{aligned} \tag{8.43}$$

The vectors $\mathscr{R}_{1,i}^{(l)}$ and $\mathscr{R}_{2,i}^{(l)}$ are defined as

$$\mathscr{R}_{1,i}^{(l)} = \mathbf{R}_{1,i}^{l} - \left(\boldsymbol{\beta}_r^{(l,1)} d_{i,N_\zeta} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,N_\zeta} \mathbf{D} \right) \mathbf{F}_{1,N_\zeta}^{(l)},$$

$$\mathscr{R}_{2,i}^{(l)} = \mathbf{R}_{2,i}^{l} - \left(\boldsymbol{\beta}_r^{(l,2)} d_{i,N_\zeta} \mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,N_\zeta} \mathbf{D} \right) \mathbf{F}_{2,N_\zeta}^{(l)}.$$

for $i = 0,1,\cdots,N_\zeta - 1$. Equations (8.42) and (8.43) are then solved iteratively for $\mathbf{F}_{k,v}^{(l)}$ for $k = 1,2$ and $v = 0,1,2,\ldots,N_\zeta - 1$. For the MD-BSQLM, we have

$$\sum_{v=1}^{2} \left[A_{1,v}^{(i,l)} \mathbf{F}_{v,i}^{(l)} + \boldsymbol{\beta}_{v,r}^{(l,1)} \sum_{j=0}^{N_\zeta} d_{i,j} \mathbf{F}_{v,j}^{(l)} + \boldsymbol{\gamma}_{v,r}^{(l,1)} \sum_{j=0}^{N_\zeta} d_{i,j} \mathbf{D} \mathbf{F}_{v,j}^{(l)} \right] = \mathbf{R}_{1,i}^{(l)}, \tag{8.44}$$

$$\sum_{v=1}^{2} \left[A_{2,v}^{(i,l)} \mathbf{F}_{v,i}^{(l)} + \boldsymbol{\beta}_{v,r}^{(l,2)} \sum_{j=0}^{N_\zeta} d_{i,j} \mathbf{F}_{v,j}^{(l)} + \boldsymbol{\gamma}_{v,r}^{(l,2)} \sum_{j=0}^{N_\zeta} d_{i,j} \mathbf{D} \mathbf{F}_{v,j}^{(l)} \right] = \mathbf{R}_{2,i}^{(l)} \tag{8.45}$$

where

$$
\begin{aligned}
A_{1,1}^{(i,l)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{1,s,r}^{(l,1)} \mathbf{D}^{(s)} = \boldsymbol{\alpha}_{1,3,r}^{(l,1)} \mathbf{D}^{(3)} + \boldsymbol{\alpha}_{1,2,r}^{(l,1)} \mathbf{D}^{(2)} + \boldsymbol{\alpha}_{1,1,r}^{(l,1)} \mathbf{D}^{(1)} + \boldsymbol{\alpha}_{1,0,r}^{(l,1)} \mathbf{I}, \\
A_{1,2}^{(i,l)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{2,s,r}^{(l,1)} \mathbf{D}^{(s)} = \boldsymbol{\alpha}_{2,3,r}^{(l,1)} \mathbf{D}^{(3)} + \boldsymbol{\alpha}_{2,2,r}^{(l,1)} \mathbf{D}^{(2)} + \boldsymbol{\alpha}_{2,1,r}^{(l,1)} \mathbf{D}^{(1)} + \boldsymbol{\alpha}_{2,0,r}^{(l,1)} \mathbf{I},
\end{aligned} \tag{8.46}
$$

$$
\begin{aligned}
A_{2,1}^{(i,l)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{1,s,r}^{(l,2)} \mathbf{D}^{(s)} = \boldsymbol{\alpha}_{1,3,r}^{(l,2)} \mathbf{D}^{(3)} + \boldsymbol{\alpha}_{1,2,r}^{(l,2)} \mathbf{D}^{(2)} + \boldsymbol{\alpha}_{1,1,r}^{(l,2)} \mathbf{D}^{(1)} + \boldsymbol{\alpha}_{1,0,r}^{(l,2)} \mathbf{I}, \\
A_{2,2}^{(i,l)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{2,s,r}^{(l,2)} \mathbf{D}^{(s)} = \boldsymbol{\alpha}_{2,3,r}^{(l,2)} \mathbf{D}^{(3)} + \boldsymbol{\alpha}_{2,2,r}^{(l,2)} \mathbf{D}^{(2)} + \boldsymbol{\alpha}_{2,1,r}^{(l,2)} \mathbf{D}^{(1)} + \boldsymbol{\alpha}_{2,0,r}^{(l,2)} \mathbf{I}.
\end{aligned} \tag{8.47}
$$

The variable coefficients for equation (8.46) are given by,

$$
\begin{aligned}
&\alpha_{1,3,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{1,r}^{(3,l)}} = 1, \qquad \alpha_{1,2,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{1,r}^{(2,l)}} = \zeta + \frac{1}{4}(n+7) f_{1,r}^{(0,l)} + \frac{1}{4}(1-n)\zeta \frac{\partial f_{1,r}^{(0,l)}}{\partial \zeta}, \\
&\alpha_{1,1,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{1,r}^{(1,l)}} = -(n+1) f_{1,r}^{(1,l)} - \frac{1}{4}(1-n)\zeta \frac{\partial f_{1,r}^{(1,l)}}{\partial \zeta}, \qquad \alpha_{1,0,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{1,r}^{(0,l)}} = \frac{1}{4}(n+7) f_{1,r}^{(2,l)}, \\
&\alpha_{2,3,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{2,r}^{(3,l)}} = 0, \qquad \alpha_{2,2,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{2,r}^{(2,l)}} = 0, \qquad \alpha_{2,1,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{2,r}^{(1,l)}} = 0, \qquad \alpha_{2,0,r}^{(l,1)} = \frac{\partial \Phi_1}{\partial f_{2,r}^{(0,l)}} = 1.
\end{aligned}
$$

The variable coefficients for equation (8.47) are given by,

$$
\begin{aligned}
&\alpha_{1,3,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{1,r}^{(3,l)}} = 0, \qquad \alpha_{1,2,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{1,r}^{(2,l)}} = 0, \qquad \alpha_{1,1,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{1,r}^{(1,l)}} = -n f_{2,r}^{(0,l)} - \frac{1}{4}(1-n)\zeta \frac{\partial f_{2,r}^{(0,l)}}{\partial \zeta}, \\
&\alpha_{1,0,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{1,r}^{(0,l)}} = \frac{1}{4}(n+7) f_{2,r}^{(1,l)}, \qquad \alpha_{2,3,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{2,r}^{(3,l)}} = 0, \qquad \alpha_{2,2,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{2,r}^{(2,l)}} = \frac{1}{Pr}, \\
&\alpha_{2,1,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{2,r}^{(1,l)}} = \frac{1}{4}(n+7) f_{1,r}^{(0,l)} + \frac{1}{4}(1-n)\zeta \frac{\partial f_{1,r}^{(0,l)}}{\partial \zeta} \qquad \alpha_{2,0,r}^{(l,2)} = \frac{\partial \Phi_2}{\partial f_{2,r}^{(0,l)}} = -n f_{1,r}^{(1,l)}.
\end{aligned}
$$

These matrices in conjunction with equation (8.16), are used to solve equations (8.31) and equation (8.32).

Steady free convective flow of a viscous incompressible fluid along a permeable vertical flat plate

A two-dimensional steady free convective flow of a viscous incompressible fluid along a permeable vertical flat plate in the presence of soluble species is being considered in this subsection. The problem considered here is that of a natural convection boundary layer flow, influenced by the combined buoyancy forces from mass and thermal diffusion from a permeable vertical flat surface with non-uniform surface temperature and non-uniform surface species concentration, but with a uniform rate of suction of fluid through the permeable surface. The transformed boundary layer equations are solved numerically near to and far from the leading edge, using the multi-domain bivariate spectral collocation method. The equations were solved by Hussain [157] and are;

$$f'''+\frac{1}{4}(n+3)ff''-\frac{1}{2}(n+1)f'^2+\zeta f''+(1-w)g+wh=\frac{1}{4}(1-n)\zeta\left[f'\frac{\partial f'}{\partial\zeta}-f''\frac{\partial f}{\partial\zeta}\right] \tag{8.48}$$

$$\frac{1}{Pr}g''+\frac{1}{4}(n+3)fg'+\zeta g'=\frac{1}{4}(1-n)\zeta\left[f'\frac{\partial g}{\partial\zeta}-g'\frac{\partial f}{\partial\zeta}\right] \tag{8.49}$$

$$\frac{1}{Sc}h''+\frac{1}{4}(n+3)fh'+\zeta h'=\frac{1}{4}(1-n)\zeta\left[f'\frac{\partial h}{\partial\zeta}-h'\frac{\partial f}{\partial\zeta}\right] \tag{8.50}$$

The appropriate boundary conditions are given by

$$\begin{aligned} f(\zeta,0)=0,\quad f'(\zeta,0)=0,\quad g(\zeta,0)=h(\zeta,0)=1, \\ f'(\zeta,\infty)=g(\zeta,\infty)=h(\zeta,\infty)=0. \end{aligned} \tag{8.51}$$

In this case, the order of differentiation $p=3$ and the number of system of equations $n=3$. For the MD-BSLLM, we have

$$A_{1,1}^{(l)}\mathbf{F}_{1,i}^{(l)}+\boldsymbol{\beta}_r^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{1,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,1)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{1,j}^{(l)}=\mathbf{R}_{1,i}^{l}, \tag{8.52}$$

$$A_{2,2}^{(l)}\mathbf{F}_{2,i}^{(l)}+\boldsymbol{\beta}_r^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{2,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,2)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{2,j}^{(l)}=\mathbf{R}_{2,i}^{l}, \tag{8.53}$$

$$A_{3,3}^{(l)}\mathbf{F}_{3,i}^{(l)}+\boldsymbol{\beta}_r^{(l,3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{F}_{3,j}^{(l)}+\boldsymbol{\gamma}_r^{(l,3)}\sum_{j=0}^{N_\zeta}d_{i,j}\mathbf{DF}_{3,j}^{(l)}=\mathbf{R}_{3,i}^{l}, \tag{8.54}$$

where

$$A_{1,1}^{(l)}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(l,1)}\mathbf{D}^{(s)}=\boldsymbol{\alpha}_{3,r}^{(l,1)}\mathbf{D}^{(3)}+\boldsymbol{\alpha}_{2,r}^{(l,1)}\mathbf{D}^{(2)}+\boldsymbol{\alpha}_{1,r}^{(l,1)}\mathbf{D}^{(1)}+\boldsymbol{\alpha}_{0,r}^{(l,1)}, \tag{8.55}$$

$$A_{2,2}^{(l)}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(l,2)}\mathbf{D}^{(s)}=\boldsymbol{\alpha}_{3,r}^{(l,2)}\mathbf{D}^{(3)}+\boldsymbol{\alpha}_{2,r}^{(l,2)}\mathbf{D}^{(2)}+\boldsymbol{\alpha}_{1,r}^{(l,2)}\mathbf{D}^{(1)}+\boldsymbol{\alpha}_{0,r}^{(l,2)}, \tag{8.56}$$

$$A_{3,3}^{(l)}=\sum_{s=0}^{3}\boldsymbol{\alpha}_{s,r}^{(l,3)}\mathbf{D}^{(s)}=\boldsymbol{\alpha}_{3,r}^{(l,3)}\mathbf{D}^{(3)}+\boldsymbol{\alpha}_{2,r}^{(l,3)}\mathbf{D}^{(2)}+\boldsymbol{\alpha}_{1,r}^{(l,3)}\mathbf{D}^{(1)}+\boldsymbol{\alpha}_{0,r}^{(l,3)}. \tag{8.57}$$

The system of equations is in terms of the functions, $f(\eta,\zeta)$, $g(\eta,\zeta)$ and $h(\eta,\zeta)$ and in order to apply our method, we set $f(\eta,\zeta)=f_1(\eta,\zeta)$, $g(\eta,\zeta)=f_2(\eta,\zeta)$ and $h(\eta,\zeta)=f_3(\eta,\zeta)$. This leads to

$$\begin{aligned}\Phi_1=&f_{1,r}^{(3,l)}+\frac{1}{4}(n+3)f_{1,r}^{(0,l)}f_{1,r}^{(2,l)}-\frac{1}{2}(n+1)\left(f_{1,r}^{(1,l)}\right)^2+\zeta f_{1,r}^{(2,l)}+(1-w)f_{2,r}^{(0,l)}+wf_{3,r}^{(0,l)}\\&-\frac{1}{4}(1-n)\zeta\left[f_{1,r}^{(1,l)}\frac{\partial f_{1,r}^{(1,l)}}{\partial\zeta}-f_{1,r}^{(2,l)}\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right]\end{aligned} \tag{8.58}$$

$$\Phi_2=\frac{1}{Pr}f_{2,r}^{(2,l)}+\frac{1}{4}(n+3)f_{1,r}^{(0,l)}f_{2,r}^{(1,l)}+\zeta f_{2,r}^{(1,l)}-\frac{1}{4}(1-n)\zeta\left[f_{1,r}^{(1,l)}\frac{\partial f_{2,r}^{(0,l)}}{\partial\zeta}-f_{2,r}^{(1,l)}\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right] \tag{8.59}$$

$$\Phi_3=\frac{1}{Sc}f_{3,r}^{(2,l)}+\frac{1}{4}(n+3)f_{1,r}^{(0,l)}f_{3,r}^{(1,l)}+\zeta f_{3,r}^{(1,l)}-\frac{1}{4}(1-n)\zeta\left[f_{1,r}^{(1,l)}\frac{\partial f_{3,r}^{(0,l)}}{\partial\zeta}-f_{3,r}^{(1,l)}\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right] \tag{8.60}$$

The variable coefficients for (8.55) are given by,

$$\alpha_{3,r}^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial f_{1,r}^{(3,l)}}=1,\qquad\alpha_{2,r}^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial f_{1,r}^{(2,l)}}=\frac{1}{4}(n+3)f_{1,r}^{(0,l)}+\zeta+\frac{1}{4}(1-n)\zeta\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta},$$

$$\alpha_{1,r}^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial f_{1,r}^{(1,l)}}=-(n+1)f_{1,r}^{(1,l)}-\frac{1}{4}(1-n)\zeta\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta},\qquad\alpha_{0,r}^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial f_{1,r}^{(0,l)}}=\frac{1}{4}(n+3)f_{1,r}^{(2,l)},$$

$$\beta_r^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial\left(\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta}\right)}=\frac{1}{4}(1-n)\zeta f_{1,r}^{(2,l)},\qquad\gamma_r^{(l,1)}(\eta,\zeta)=\frac{\partial\Phi_1}{\partial\left(\frac{\partial f_{1,r}^{(1,l)}}{\partial\zeta}\right)}=-\frac{1}{4}(1-n)\zeta f_{1,r}^{(1,l)}.$$

for (8.56) are given by,

$$\gamma_r^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial\left(\frac{\partial f_{2,r}^{(1,l)}}{\partial\zeta}\right)}=0,\qquad \alpha_{3,r}^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial f_{2,r}^{(3,l)}}=0,\qquad \alpha_{2,r}^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial f_{2,r}^{(2,l)}}=\frac{1}{Pr},$$

$$\alpha_{1,r}^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial f_{2,r}^{(1,l)}}=\frac{1}{4}(n+3)f_{1,r}^{(0,l)}+\zeta+\frac{1}{4}(1-n)\zeta\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta},\qquad \alpha_{0,r}^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial f_{2,r}^{(0,l)}}=0,$$

$$\beta_r^{(l,2)}(\eta,\zeta)=\frac{\partial\Phi_2}{\partial\left(\frac{\partial f_{2,r}^{(0,l)}}{\partial\zeta}\right)}=-\frac{1}{4}(1-n)\zeta f_{1,r}^{(1,l)}.$$

and for equation (8.57) are given by,

$$\gamma_r^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial\left(\frac{\partial f_{3,r}^{(1,l)}}{\partial\zeta}\right)}=0,\qquad \alpha_{3,r}^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial f_{3,r}^{(3,l)}}=0,\qquad \alpha_{2,r}^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial f_{3,r}^{(2,l)}}=\frac{1}{Sc},$$

$$\alpha_{1,r}^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial f_{3,r}^{(1,l)}}=\frac{1}{4}(n+3)f_{1,r}^{(0,l)}+\zeta+\frac{1}{4}(1-n)\zeta\frac{\partial f_{1,r}^{(0,l)}}{\partial\zeta},\qquad \alpha_{0,r}^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial f_{3,r}^{(0,l)}}=0,$$

$$\beta_r^{(l,3)}(\eta,\zeta)=\frac{\partial\Phi_3}{\partial\left(\frac{\partial f_{3,r}^{(0,l)}}{\partial\zeta}\right)}=-\frac{1}{4}(1-n)\zeta f_{1,r}^{(1,l)}.$$

Equations (8.52), (8.53) and (8.54) can be expressed respectively as the following $N_\zeta(N_\eta+1)\times N_\zeta(N_\eta+1)$ matrix system

$$\begin{bmatrix} B_{0,0}^{(l,1)} & B_{0,1}^{(l,1)} & \cdots & B_{0,N_\zeta-1}^{(l,1)} \\ B_{1,0}^{(l,1)} & B_{1,1}^{(l,1)} & \cdots & B_{1,N_\zeta-1}^{(l,1)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,1)} & B_{N_\zeta-1,1}^{(l,1)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,1)} \end{bmatrix}\begin{bmatrix} \mathbf{F}_{1,0}^{(l)} \\ \mathbf{F}_{1,1}^{(l)} \\ \vdots \\ \mathbf{F}_{1,N_\zeta-1}^{(l)} \end{bmatrix}=\begin{bmatrix} \mathscr{R}_{1,0}^{(l)} \\ \mathscr{R}_{1,1}^{(l)} \\ \vdots \\ \mathscr{R}_{1,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.61}$$

$$\begin{bmatrix} B_{0,0}^{(l,2)} & B_{0,1}^{(l,2)} & \cdots & B_{0,N_\zeta-1}^{(l,2)} \\ B_{1,0}^{(l,2)} & B_{1,1}^{(l,2)} & \cdots & B_{1,N_\zeta-1}^{(l,2)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,2)} & B_{N_\zeta-1,1}^{(l,2)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,2)} \end{bmatrix}\begin{bmatrix} \mathbf{F}_{2,0}^{(l)} \\ \mathbf{F}_{2,1}^{(l)} \\ \vdots \\ \mathbf{F}_{2,N_\zeta-1}^{(l)} \end{bmatrix}=\begin{bmatrix} \mathscr{R}_{2,0}^{(l)} \\ \mathscr{R}_{2,1}^{(l)} \\ \vdots \\ \mathscr{R}_{2,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.62}$$

$$\begin{bmatrix} B_{0,0}^{(l,3)} & B_{0,1}^{(l,3)} & \cdots & B_{0,N_\zeta-1}^{(l,3)} \\ B_{1,0}^{(l,3)} & B_{1,1}^{(l,3)} & \cdots & B_{1,N_\zeta-1}^{(l,3)} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N_\zeta-1,0}^{(l,3)} & B_{N_\zeta-1,1}^{(l,3)} & \cdots & B_{N_\zeta-1,N_\zeta-1}^{(l,3)} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{3,0}^{(l)} \\ \mathbf{F}_{3,1}^{(l)} \\ \vdots \\ \mathbf{F}_{3,N_\zeta-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathscr{R}_{3,0}^{(l)} \\ \mathscr{R}_{3,1}^{(l)} \\ \vdots \\ \mathscr{R}_{3,N_\zeta-1}^{(l)} \end{bmatrix}, \tag{8.63}$$

where

$$\begin{aligned} B_{(i,i)}^{(l,1)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(l,1)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,1)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,i}\mathbf{D}, \quad \text{for } l = 1,2,\ldots,z \text{ when } i = j, \\ B_{(i,j)}^{(l,1)} &= \boldsymbol{\beta}_r^{(l,1)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,j}\mathbf{D}, \quad \text{for, } l = 1,2,\ldots,z \text{ when } i \neq j, \end{aligned} \tag{8.64}$$

$$\begin{aligned} B_{(i,i)}^{(l,2)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(l,2)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,2)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,i}\mathbf{D}, \quad \text{for } l = 1,2,\ldots,z \text{ when } i = j, \\ B_{(i,j)}^{(l,2)} &= \boldsymbol{\beta}_r^{(l,2)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,j}\mathbf{D}, \quad \text{for, } l = 1,2,\ldots,z \text{ when } i \neq j, \end{aligned} \tag{8.65}$$

$$\begin{aligned} B_{(i,i)}^{(l,3)} &= \sum_{s=0}^{3} \boldsymbol{\alpha}_{s,r}^{(l,3)} \mathbf{D}^{(s)} + \boldsymbol{\beta}_r^{(l,3)} d_{i,i}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,3)} d_{i,i}\mathbf{D}, \quad \text{for } l = 1,2,\ldots,z \text{ when } i = j, \\ B_{(i,j)}^{(l,3)} &= \boldsymbol{\beta}_r^{(l,3)} d_{i,j}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,3)} d_{i,j}\mathbf{D}, \quad \text{for, } l = 1,2,\ldots,z \text{ when } i \neq j. \end{aligned} \tag{8.66}$$

The vectors $\mathscr{R}_{1,i}^{(l)}$, $\mathscr{R}_{2,i}^{(l)}$ and $\mathscr{R}_{3,i}^{(l)}$ are defined as

$$\mathscr{R}_{1,i}^{(l)} = \mathbf{R}_{1,i}^{l} - \left(\boldsymbol{\beta}_r^{(l,1)} d_{i,N_\zeta}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,1)} d_{i,N_\zeta}\mathbf{D}\right) \mathbf{F}_{1,N_\zeta}^{(l)},$$

$$\mathscr{R}_{2,i}^{(l)} = \mathbf{R}_{2,i}^{l} - \left(\boldsymbol{\beta}_r^{(l,2)} d_{i,N_\zeta}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,2)} d_{i,N_\zeta}\mathbf{D}\right) \mathbf{F}_{2,N_\zeta}^{(l)},$$

$$\mathscr{R}_{3,i}^{(l)} = \mathbf{R}_{3,i}^{l} - \left(\boldsymbol{\beta}_r^{(l,3)} d_{i,N_\zeta}\mathbf{I} + \boldsymbol{\gamma}_r^{(l,3)} d_{i,N_\zeta}\mathbf{D}\right) \mathbf{F}_{3,N_\zeta}^{(l)}.$$

for $i = 0,1,\cdots,N_\zeta - 1$. For the MD-BSQLM, the procedure is similar to that in the previous example.

8.5 Series solution for the limiting case

In this section, we solve the systems of equations presented above using the series solution approach. These series solutions for the limiting case when ζ is very large are used to validate the results obtained using the multi-domain bivariate spectral local linearization method. Equations of the form (8.31 - 8.32) and (8.48 - 8.50) can be decomposed to a sequence of coupled linear ordinary differential equations when $\zeta \to \infty$. The solutions of the resulting systems of equations can be determined using elementary methods for solving differential equations.

Consider the series solution of equations (8.31) - (8.32). For large values of ζ, the dominant terms in equations (8.31) are f''' and $\zeta f''$. Balancing the orders of magnitude of these terms gives $\eta = \mathscr{O}(\zeta^{-1})$. Since $\theta(0) = 1$, we may infer that $\theta(\eta,\zeta)$ is of $\mathscr{O}(1)$ as $\zeta \to \infty$. Assuming that f''' has the same order as θ for large ζ, then $f = \mathscr{O}(\zeta^{-3})$. Thus for large values of ζ, we define the following transformations;

$$\bar{\eta} = \zeta\eta, \quad f = \zeta^{-3}F(\bar{\eta},\zeta), \quad \theta = \Theta(\bar{\eta},\zeta). \tag{8.67}$$

The transformation equations (8.67) reduces equations (8.31) - (8.32) to

$$F'''+F''+\Theta+(1+n)\zeta^{-4}FF''-n\zeta^{-4}F'^2 = \frac{1-n}{4}\zeta^{-3}\left[F'\frac{\partial F'}{\partial\zeta}-F''\frac{\partial F}{\partial\zeta}\right], \tag{8.68}$$

$$\frac{1}{Pr}\Theta''+\Theta'+(1+n)\zeta^{-4}F\Theta'-n\zeta^{-4}F'\Theta = \frac{1-n}{4}\zeta^{-3}\left[F'\frac{\partial\Theta}{\partial\zeta}-\Theta'\frac{\partial F}{\partial\zeta}\right], \tag{8.69}$$

where the primes now denote derivatives with respect to $\bar{\eta}$. The corresponding boundary conditions are given by

$$F(\zeta,0) = F'(\zeta,0) = 0, \quad \Theta(\zeta,0) = 1, \quad F'(\zeta,\infty) = \Theta(\zeta,\infty) = 0. \tag{8.70}$$

Using the fact that ζ is large, we therefore seek solutions to equations (8.68) - (8.69) in series form as

$$F(\zeta,\bar{\eta}) = \sum_{k=0}^{\infty}\zeta^{-4k}F_k(\bar{\eta}), \quad \Theta(\zeta,\bar{\eta}) = \sum_{k=0}^{\infty}\zeta^{-4k}\Theta_k(\bar{\eta}). \tag{8.71}$$

Substituting equations (8.71) into equations (8.68) - (8.70) and then equating the coefficients of like powers of ζ, we obtain the equations for $k = 0$ as;

$$F_0''' + F_0'' + \Theta_0 = 0, \tag{8.72}$$

$$\frac{1}{Pr}\Theta_0'' + \Theta_0' = 0, \tag{8.73}$$

subject to the following boundary conditions

$$F_0(0) = F_0'(0) = 0, \quad \Theta_0(0) = 1, \quad F_0'(\infty) = \Theta_0(\infty) = 0. \tag{8.74}$$

The analytical solutions of the zeroth order equations (8.72) - (8.74) are;

$$F_0(\bar{\eta}) = \frac{e^{-Pr\bar{\eta}}}{(Pr-1)Pr^2} - \frac{1}{(Pr-1)Pr^2} - \frac{e^{-\bar{\eta}}}{(Pr-1)Pr} + \frac{1}{(Pr-1)Pr}, \tag{8.75}$$

$$\Theta_0(\bar{\eta}) = e^{-Pr\bar{\eta}}. \tag{8.76}$$

The following theorems simplify the multiplication of two infinite series and two power series together.

Theorem 8.1. *(Cauchy product of two infinite series) Consider two infinite series* $\sum_{n=0}^{\infty} a_n$ *and* $\sum_{n=0}^{\infty} b_n$. *Then whenever both of these infinite series are convergent, we have*

$$\left(\sum_{i=0}^{\infty} a_i\right)\left(\sum_{j=0}^{\infty} b_j\right) = \sum_{k=0}^{\infty}\left(\sum_{l=0}^{k} a_l b_{k-l}\right). \tag{8.77}$$

Theorem 8.2. *(Cauchy product of two power series) Consider two power series* $\sum_{n=0}^{\infty} a_n\zeta^n$ *and* $\sum_{n=0}^{\infty} b_n\zeta^n$. *Then whenever both of these power series are convergent, we have*

$$\left(\sum_{i=0}^{\infty} a_i\zeta^i\right)\left(\sum_{j=0}^{\infty} b_j\zeta^j\right) = \sum_{k=0}^{\infty}\left[\left(\sum_{l=0}^{k} a_l b_{k-l}\right)\zeta^k\right]. \tag{8.78}$$

Thus, for $k \geq 1$, we sequentially solve the following system of ordinary differential equations;

$$F_k''' + F_k'' + \Theta_k = n\sum_{i=0}^{k-1} F'_{k-1-i}F'_i - (1+n)\sum_{i=0}^{k-1} F_{k-1-i}F_i'' + (1-n)\left[\sum_{i=0}^{k-1} i\left(F''_{k-1-i}F_i - F'_{k-1-i}F'_i\right)\right], \quad (8.79)$$

$$\frac{1}{Pr}\Theta_k'' + \Theta_k' = n\sum_{i=0}^{k-1} F'_{k-1-i}\Theta_i - (1+n)\sum_{i=0}^{k-1} F_{k-1-i}\Theta_i' + (1-n)\left[\sum_{i=0}^{k-1} i\left(\Theta'_{k-1-i}F_i - F'_{k-1-i}\Theta_i\right)\right], \quad (8.80)$$

subject to the following boundary conditions

$$F_k(0) = F_k'(0) = 0, \quad \Theta_k(0) = 0, \quad F_k'(\infty) = \Theta_k(\infty) = 0. \quad (8.81)$$

We now consider the series solution of equations (8.48) - (8.50). For large values of ζ, the dominant terms in equation (8.48) are f''' and $\zeta f''$. Balancing terms in equations (8.48) - (8.50), for $\zeta \to \infty$, we get

$$\eta = \mathscr{O}(\zeta^{-1}), \quad f = \mathscr{O}(\zeta^{-3}), \quad g = \mathscr{O}(1), \quad h = \mathscr{O}(1). \quad (8.82)$$

This leads to the following transformations

$$f = \zeta^{-3}F(\eta,\zeta), \quad \bar{\eta} = \zeta\eta, \quad g = G(\eta,\zeta), \quad h = H(\eta,\zeta). \quad (8.83)$$

Using the transformations (8.82) - (8.83) in (8.48) - (8.50), we obtain

$$F''' + F'' + (1-w)G + wH + n\zeta^{-4}\left(FF'' - F'^2\right) = \frac{1-n}{4}\zeta^{-3}\left[F'\frac{\partial F'}{\partial\zeta} - F''\frac{\partial F}{\partial\zeta}\right], \quad (8.84)$$

$$\frac{1}{Pr}G'' + G' + n\zeta^{-4}FG' = \frac{1-n}{4}\zeta^{-3}\left[F'\frac{\partial G}{\partial\zeta} - G'\frac{\partial F}{\partial\zeta}\right], \quad (8.85)$$

$$\frac{1}{Sc}H'' + H' + n\zeta^{-4}FH' = \frac{1-n}{4}\zeta^{-3}\left[F'\frac{\partial H}{\partial\zeta} - H'\frac{\partial F}{\partial\zeta}\right]. \quad (8.86)$$

The corresponding boundary conditions are given by

$$F(\zeta,0) = F'(\zeta,0) = 0, \quad G(\zeta,0) = 1, \quad H(\zeta,0) = 1, \quad F'(\zeta,\infty) = G(\zeta,\infty) = H(\zeta,\infty) = 0. \quad (8.87)$$

For large ζ, we seek solutions to equations (8.84) - (8.87) in series form as

$$F(\zeta,\bar{\eta})=\sum_{k=0}^{\infty}\zeta^{-4k}F_k(\bar{\eta}),\quad G(\zeta,\bar{\eta})=\sum_{k=0}^{\infty}\zeta^{-4k}G_k(\bar{\eta}),\quad H(\zeta,\bar{\eta})=\sum_{k=0}^{\infty}\zeta^{-4k}H_k(\bar{\eta}). \tag{8.88}$$

Substituting equations (8.88) into equations (8.84) - (8.87) and then equating the coefficients of like powers of ζ, we obtain the equations for $k=0$ as

$$F_0'''+F_0''+(1-w)G_0+wH_0=0, \tag{8.89}$$

$$\frac{1}{Pr}G_0''+G_0'=0, \tag{8.90}$$

$$\frac{1}{Sc}H_0''+H_0'=0, \tag{8.91}$$

subject to the following boundary conditions

$$F_0(0)=F_0'(0)=0,\quad G_0(0)=1,\quad H_0(0)=1,\quad F_0'(\infty)=G_0(\infty)=H_0(\infty)=0. \tag{8.92}$$

The sequence of equations obtained when $k\geq 1$ are given as

$$\begin{aligned}
&F_k'''+F_k''+(1-w)G_k+wH_k=n\sum_{i=0}^{k-1}\left(F_{k-1-i}'F_i'-F_{k-1-i}F_i''\right)+(1-n)\left[\sum_{i=0}^{k-1}i\left(F_{k-1-i}''F_i-F_{k-1-i}'F_i'\right)\right],\\
&\frac{1}{Pr}G_k''+G_k'=-n\sum_{i=0}^{k-1}F_{k-1-i}G_i'+(1-n)\left[\sum_{i=0}^{k-1}i\left(G_{k-1-i}'F_i-F_{k-1-i}'G_i\right)\right],\\
&\frac{1}{Sc}H_k''+H_k'=-n\sum_{i=0}^{k-1}F_{k-1-i}H_i'+(1-n)\left[\sum_{i=0}^{k-1}i\left(H_{k-1-i}'F_i-F_{k-1-i}'H_i\right)\right],
\end{aligned} \tag{8.93}$$

subject to the following boundary conditions

$$F_k(0)=F_k'(0)=0,\quad G_k(0)=1,\quad H_k(0)=1,\quad F_k'(\infty)=G_k(\infty)=H_k(\infty)=0. \tag{8.94}$$

8.6 Results and Discussion

In this section, we present and analyze the results obtained using the multi-domain bivariate spectral local linearisation method (MD-BSLLM), multi-domain bivariate spectral quasilinearisation method (MD-BSQLM) and the series solution. The MD-BSLLM and MD-BSQLM were

implemented using MATLAB 2013b. The series solutions were obtained using Mathematica's `NDSolve`. The parameters to generate all the solutions are given in captions for the tables and graphs. In this work, the physical quantities of interest in the field of fluid dynamics, the skin friction, Sherwood number and Nusselt number are presented. These physical quantities were obtained using all methods presented here. We present the values of these quantities for different large values of ζ. The values of these quantities are in excellent agreement for all the methods presented in this work. This in turn implies that the new approaches (MD-BSLLM, MD-BSQLM) presented here can be used to solve other non-similar boundary layer partial differential equations. The effect of number of intervals in the solution is also presented in this section. Graphs showing the velocity profiles, temperature profiles and concentration profiles are also presented for various large values of ζ. These profiles are in excellent agreement with the results presented by Hussain [157].

We first present the solutions of equations (8.31) and (8.32). The physical quantities, skin friction and the Nusselt number results obtained by solving the governing equations (8.31) - (8.32) are given in Table 8.1. These results were obtained directly by solving the sequences of ordinary differential equations, for the large ζ limiting case, using Mathematica's `DSolve` command. The series solutions, MD-BSQLM and the MD-BSLLM solutions are accurate up to at least eight decimal digits as indicated in Table 8.1. Both solutions are in excellent agreement and thus validating the accuracy of the MD-BSQLM and MD-BSLLM methods. We observe that an increase in ζ results in a decrease in both the skin friction and the Nusselt number. It is remarkable that this numerical methods were able to produce accurate results for large values of ζ with minimum number of grid points. These results were obtained using $N_\zeta = 5$, $N_\eta = 60$ and converged after five iterations.

Table 8.1 Comparison of the MD-BSLLM solution, MD-BSQLM solution, and the series solution for $f''(0,\zeta)$ and $\theta'(0,\zeta)$, with $Pr = 0.7$, $n = 1/2$ and $q = 40$.

	Multi-Domain BSLLM		Multi-Domain BSQLM		Series Solution	
ζ	$f''(0,\zeta)$	$\theta'(0,\zeta)$	$f''(0,\zeta)$	$\theta'(0,\zeta)$	$f''(0,\zeta)$	$\theta'(0,\zeta)$
5	0.2842719	-3.5066677	0.2842719	-3.5066677	0.2842719	-3.5066677
10	0.1428115	-7.0008399	0.1428115	-7.0008399	0.1428115	-7.0008399
15	0.0952321	-10.5002490	0.0952321	-10.5002490	0.09523208	-10.5002490
20	0.0714271	-14.0001050	0.0714271	-14.0001050	0.07142714	-14.0001050
25	0.0571424	-17.5000538	0.0571424	-17.5000538	0.05714239	-17.5000538
30	0.0476189	-21.0000311	0.0476189	-21.0000311	0.04761886	-21.0000311
35	0.0408162	-24.5000196	0.0408162	-24.5000196	0.04081624	-24.5000196
40	0.0357142	-28.0000131	0.0357142	-28.0000131	0.03571424	-28.0000131

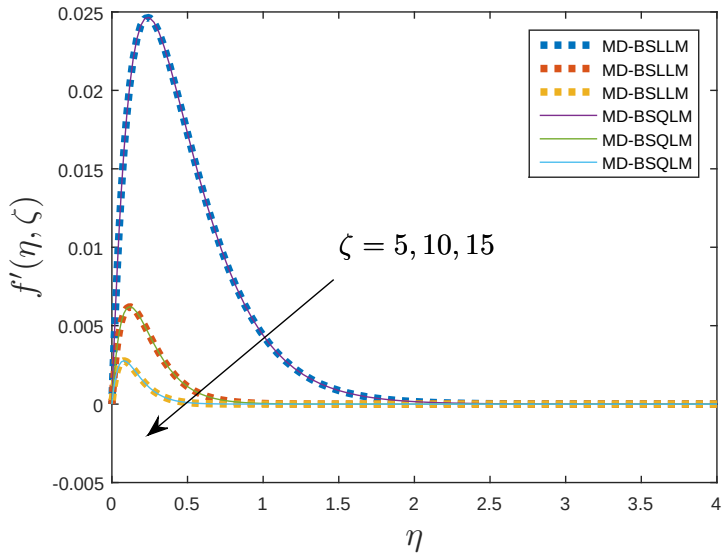

Fig. 8.1 Velocity profiles at different values of ζ with $Pr = 0.7$, $n = 1/2$ and $p = 40$.

Figure 8.1 shows the velocity profiles for different values of ζ. An increase in ζ leads to a decrease in the velocity. The velocity profile is parabolic is in excellent agreement with that obtained by Hussain [157], thus the MD-BSLLM and MD-BSLLM methods can be used as numerical method for solving large parameter partial differential equations. The MD-BSLLM and MD-BSLLM methods produce similar results and hence validating the accuracy of each other.

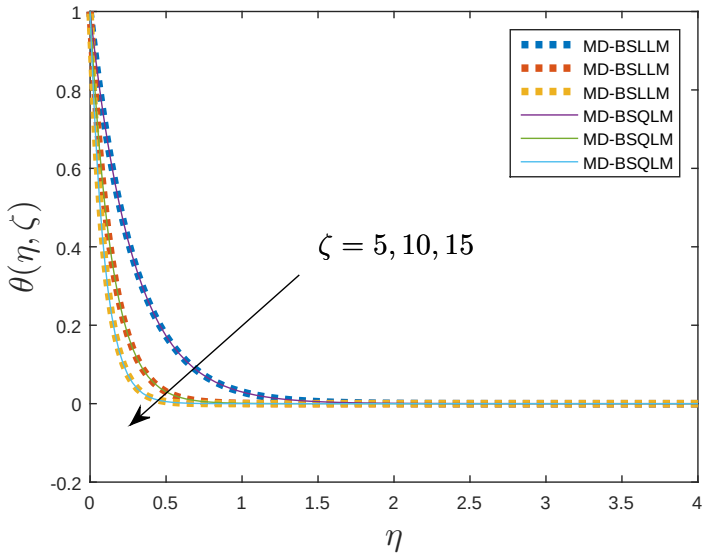

Fig. 8.2 Temperature profiles at different values of ζ with $Pr = 0.7$, $n = 1/2$ and $q = 40$.

Figure 8.2 shows the temperature profile for various values of ζ. The results were obtained using the MD-BSLLM and MD-BSLLM methods. The temperature profile behaves like an exponential decay curve, that is, an increase in the pseudo-similarity variable η, results in a decrease in the temperature. Temperature profiles were also obtained for large values of ζ. Decreasing ζ leads to an increase in the temperature profile. Figures 8.1 and 8.2 were obtained using $N_\zeta = 5$, and $N_\eta = 60$.

For the second numerical experiment, the skin friction, Sherwood number and Nusselt number results obtained by solving the governing equations (8.48) - (8.50) are given in Table 8.2. Similarly, these results were obtained directly by solving the sequences of ordinary differential equations, for the large ζ, using Mathematica's `DSolve` command. The series solutions and the MD-BSLLM solutions are accurate up to at least eight decimal digits as indicated in Table 8.2. The MD-BSLLM and series solutions of equations (8.48) - (8.50) are in excellent agreement and thus validating the accuracy of the MD-BSLLM method. Increasing the transpiration parameter ζ results in a decrease of the skin friction, Sherwood number and Nusselt number. These results were obtained using few grid points, $N_\zeta = 5$ and $N_\eta = 60$. Similar observation is made for Table 8.3 where the MD-BSQLM method was used to solve Table 8.2.

Table 8.2 Comparison of the Multi-domain solution and the series solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$:$n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$.

	Multi-Domain BSLLM			Series Solution		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3088214	3.5018961	3.0018658	0.3088214	3.5018961	3.0018658
10	0.1547399	7.0002370	6.0002332	0.1547399	7.0002370	6.0002332
15	0.1031717	10.5000702	9.0000691	0.1031717	10.5000702	9.0000691
20	0.0773803	14.0000296	12.0000292	0.0773803	14.0000296	12.0000292
25	0.0619045	17.5000152	15.0000149	0.0619045	17.5000152	15.0000149
30	0.0515872	21.0000088	18.0000086	0.0515872	21.0000088	18.0000086
35	0.0442176	24.5000055	21.0000054	0.0442176	24.5000055	21.0000054
40	0.0386905	28.0000037	24.0000036	0.0386905	28.0000037	24.0000036

Table 8.3 Comparison of the Multi-domain solution and the series solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$:$n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$.

	Multi-Domain BSQLM			Series Solution		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3088214	3.5018961	3.0018658	0.3088214	3.5018961	3.0018658
10	0.1547399	7.0002370	6.0002332	0.1547399	7.0002370	6.0002332
15	0.1031717	10.5000702	9.0000691	0.1031717	10.5000702	9.0000691
20	0.0773803	14.0000296	12.0000292	0.0773803	14.0000296	12.0000292
25	0.0619045	17.5000152	15.0000149	0.0619045	17.5000152	15.0000149
30	0.0515872	21.0000088	18.0000086	0.0515872	21.0000088	18.0000086
35	0.0442176	24.5000055	21.0000054	0.0442176	24.5000055	21.0000054
40	0.0386905	28.0000037	24.0000036	0.0386905	28.0000037	24.0000036

Table 8.4 gives a comparison of the results obtained using the MD-BSLLM and the bivariate spectral local linearisation method (BSLLM) for large values of ζ. The main purpose of the results is to confirm that the accuracy of the BSLLM method deteriorates as ζ increases. We observe that the computed values agree up to an average of three digits. This implies that decomposing the main interval into subintervals and solving the non-similar partial differential

equations in each sub-interval improves the accuracy of the BSLLM method. Table 8.4 in addition displays the computational time of both methods. For $N_\eta = 60$, $N_\zeta = 5$ and $q = 40$, the BSLLM method takes about a fraction of a second to achieve inaccurate results meanwhile the MD-BSLLM method takes about 18 seconds to achieve accurate results.

Table 8.4 Comparison of the MD-BSLLM and the BSLLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	BSLLM			Multi-Domain BSLLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3046603	3.5051984	3.0093278	0.3088214	3.5018961	3.0018658
10	0.1546097	7.0003327	6.0003186	0.1547399	7.0002370	6.0002332
15	0.1030889	10.5001437	9.0001125	0.1031717	10.5000702	9.0000691
20	0.0773803	14.0000294	12.0000289	0.0773803	14.0000296	12.0000292
25	0.0619452	17.4999716	14.9999874	0.0619045	17.5000152	15.0000149
30	0.0516135	20.9999793	17.9999897	0.0515872	21.0000088	18.0000086
CPU Time	0.845	0.845	0.845	17.527	17.527	17.527

Table 8.5 gives a comparison of the results obtained using the MD-BSQLM and the bivariate spectral quasilinearisation method (BSQLM) for large values of ζ. The main purpose of the results is to confirm that the accuracy of the BSQLM method deteriorates as ζ increases. We observe that the computed values agree up to an average of three digits. This implies that decomposing the main interval into subintervals and solving the non-similar partial differential equations in each sub-interval improves the accuracy of the BSQLM method. Table 8.5 in addition displays the computational time of both methods. For $N_\eta = 60$, $N_\zeta = 5$ and $q = 40$, the BSQLM method takes about a fraction of a second to achieve inaccurate results meanwhile the MD-BSQLM method takes about 13 seconds to achieve accurate results.

Table 8.5 Comparison of the MD-BSQLM and the BSQLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	BSQLM			Multi-Domain BSQLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3046603	3.5051984	3.0093278	0.3088214	3.5018961	3.0018658
10	0.1546097	7.0003327	6.0003186	0.1547399	7.0002370	6.0002332
15	0.1030889	10.5001437	9.0001125	0.1031717	10.5000702	9.0000691
20	0.0773803	14.0000294	12.0000289	0.0773803	14.0000296	12.0000292
25	0.0619452	17.4999716	14.9999874	0.0619045	17.5000152	15.0000149
30	0.0516135	20.9999793	17.9999897	0.0515872	21.0000088	18.0000086
CPU Time	0.545	0.545	0.545	12.527	12.527	12.527

Table 8.6 shows a comparison of the BSLLM method and the MD-BSLLM method with different grid points. For the MD-BSLLM method, we use $N_\eta = 60$, $N_\zeta = 5$ and $q = 40$, while for the BSLLM method, we use $N_\eta = 40$, and $N_\zeta = 80$ to achieve comparable accurate results. Since the BSLLM method requires more grid points to achieve accurate results over a large domain, then it takes more computational time compared to the MD-BSLLM method over the same interval. In Table 8.6, the MD-BSLLM method takes about 18 seconds to achieve accurate results meanwhile the BSLLM method takes about 78 seconds to achieve accurate results. This implies that the MD-BSLLM method takes less computational time compared to the BSLLM method to achieve accurate results.

Table 8.6 Comparison of the MD-BSLLM and the BSLLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	BSLLM			Multi-Domain BSLLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3088066	3.5018993	3.0018843	0.3088214	3.5018961	3.0018658
10	0.1547399	7.0002370	6.0002332	0.1547399	7.0002370	6.0002332
15	0.1031717	10.5000702	9.0000691	0.1031717	10.5000702	9.0000691
20	0.0773801	14.0000296	12.0000292	0.0773803	14.0000296	12.0000292
25	0.0619014	17.5000152	15.0000149	0.0619045	17.5000152	15.0000149
30	0.0515660	21.0000087	18.0000086	0.0515872	21.0000088	18.0000086
CPU Time	77.921	77.921	77.921	17.527	17.527	17.527

Table 8.7 shows a comparison of the BSQLM method and the MD-BSQLM method with different grid points. For the MD-BSQLM method, we use $N_\eta = 60$, $N_\zeta = 5$ and $q = 40$, while for the BSQLM method, we use $N_\eta = 40$, and $N_\zeta = 80$ to achieve comparable accurate results. Since the BSQLM method requires more grid points to achieve accurate results over a large domain, then it takes more computational time compared to the MD-BSQLM method over the same interval. In Table 8.7, the MD-BSQLM method takes about 13 seconds to achieve accurate results meanwhile the BSQLM method takes about 72 seconds to achieve accurate results. This implies that the MD-BSQLM method takes less computational time compared to the BSQLM method to achieve accurate results.

Table 8.7 Comparison of the MD-BSQLM and the BSQLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	BSQLM			Multi-Domain BSQLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
5	0.3088066	3.5018993	3.0018843	0.3088214	3.5018961	3.0018658
10	0.1547399	7.0002370	6.0002332	0.1547399	7.0002370	6.0002332
15	0.1031717	10.5000702	9.0000691	0.1031717	10.5000702	9.0000691
20	0.0773801	14.0000296	12.0000292	0.0773803	14.0000296	12.0000292
25	0.0619014	17.5000152	15.0000149	0.0619045	17.5000152	15.0000149
30	0.0515660	21.0000087	18.0000086	0.0515872	21.0000088	18.0000086
CPU Time	71.921	71.921	71.921	12.527	12.527	12.527

In Table 8.8, we observe that for small values of ζ, the results from both methods (BSLLM and MD-BSLLM) are in good agreement. The BSLLM method takes about 3 seconds while the MD-BSLLM method takes about 22 seconds to achieve accurate results. Therefore, we can conclude that it is not necessary to use the MD-BSLLM method to solve the partial differential equations in small subdomains of ζ. The multi-domain approach is suitable for solving equations with large values of ζ. The results in Table 8.8 were obtained with $N_\eta = 60$, and $N_\zeta = 15$ for the BSLLM method meanwhile $N_\eta = 60$, $N_\zeta = 5$, and $z = 40$ was used for the MD-BSLLM method for $\zeta = 2$.

Table 8.8 Comparison of the MD-BSLLM and the BSLLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$:$n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$

	BSLLM			Multi-Domain BSLLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
0.25	0.6346269	0.6230281	0.6046615	0.6346269	0.6230281	0.6046615
0.50	0.6393410	0.7247205	0.6899358	0.6393410	0.7247205	0.6899358
0.75	0.6379544	0.8353765	0.7818927	0.6379544	0.8353765	0.7818927
1.00	0.6307080	0.9546061	0.8803437	0.6307080	0.9546061	0.8803437
1.25	0.6181647	1.0818907	0.9850250	0.6181647	1.0818907	0.9850250
1.50	0.6011394	1.2166019	1.0955986	0.6011394	1.2166019	1.0955986
1.75	0.5806016	1.3580272	1.2116586	0.5806016	1.3580272	1.2116586
2.00	0.5575697	1.5054016	1.3327435	0.5575697	1.5054016	1.3327435
CPU Time	3.088	3.088	3.088	22.436	22.436	22.436

In Table 8.9, we observe that for small values of ζ, the results from both methods (BSQLM and MD-BSQLM) are in good agreement. The BSQLM method takes about a second while the MD-BSQLM method takes about 17 seconds to achieve accurate results. Therefore, we can conclude that it is not necessary to use the MD-BSQLM method to solve the partial differential equations in small subdomains of ζ. The multi-domain approach is suitable for solving equations with large values of ζ. The results in Table 8.9 were obtained with $N_\eta = 60$, and $N_\zeta = 15$ for the BSQLM method meanwhile $N_\eta = 60$, $N_\zeta = 5$, and $q = 40$ was used for the MD-BSQLM method for $\zeta = 2$.

Table 8.9 Comparison of the MD-BSQLM and the BSQLM solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$:$n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$

	BSQLM			Multi-Domain BSQLM		
ζ	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
0.25	0.6346269	0.6230281	0.6046615	0.6346269	0.6230281	0.6046615
0.50	0.6393410	0.7247205	0.6899358	0.6393410	0.7247205	0.6899358
0.75	0.6379544	0.8353765	0.7818927	0.6379544	0.8353765	0.7818927
1.00	0.6307080	0.9546061	0.8803437	0.6307080	0.9546061	0.8803437
1.25	0.6181647	1.0818907	0.9850250	0.6181647	1.0818907	0.9850250
1.50	0.6011394	1.2166019	1.0955986	0.6011394	1.2166019	1.0955986
1.75	0.5806016	1.3580272	1.2116586	0.5806016	1.3580272	1.2116586
2.00	0.5575697	1.5054016	1.3327435	0.5575697	1.5054016	1.3327435
CPU Time	1.088	1.088	1.088	17.416	17.416	17.416

Table 8.10 shows the effect of the number of sub-intervals q. Table 8.10 shows that increasing the number of sub-intervals yields more accurate results. We note that when $z = 1$, that is we have one domain, then the MD-BSLLM is behaves exactly as the BSLLM method. For $z = 1$, the results do not match because the MD-BSLLM is exactly the BSLLM method. However, for $\zeta = 40$, only ten subintervals ensure accuracy of the MD-BSLLM method as depicted in Table 8.10. These results confirm that decomposing the main domain into subintervals improves the accuracy of the BSLLM method.

Table 8.10 Comparison of the Multi-domain solution and the series solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, η_∞, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	Multi-Domain BSLLM			Series Solution		
q	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
1	0.0967112	11.2002848	9.6003085	0.0967241	11.2000579	9.6000569
5	0.0967241	11.2000577	9.6000568	0.0967241	11.2000579	9.6000569
10	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569
15	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569
20	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569

Table 8.11 shows the effect of the number of sub-intervals q. Table 8.11 shows that increasing the number of sub-intervals yields more accurate results. We note that when $q = 1$, that is we have one domain, then the MD-BSQLM is behaves exactly as the BSQLM method. For $q = 1$, the results do not match because the MD-BSQLM is exactly the BSQLM method. However, for $\zeta = 40$, only ten subintervals ensure accuracy of the MD-BSQLM method as depicted in Table 8.11. These results confirm that decomposing the main domain into subintervals improves the accuracy of the BSQLM method.

Table 8.11 Comparison of the Multi-domain solution and the series solution for $f''(0,\zeta)$, $-g'(0,\zeta)$ and $-h'(0,\zeta)$: $N_\eta = 60$, $N_\zeta = 5$, η_∞, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$

	Multi-Domain BSQLM			Series Solution		
q	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$	$f''(0,\zeta)$	$-g'(0,\zeta)$	$-h'(0,\zeta)$
1	0.0967112	11.2002848	9.6003085	0.0967241	11.2000579	9.6000569
5	0.0967241	11.2000577	9.6000568	0.0967241	11.2000579	9.6000569
10	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569
15	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569
20	0.0967241	11.2000579	9.6000569	0.0967241	11.2000579	9.6000569

Figures 8.3, 8.4 and 8.5 show the velocity, temperature and concentration profiles for various values of ζ, respectively. It is observed that an increase in ζ leads to a decrease in the velocity, temperature and concentration. The temperature and concentration profiles decay exponentially. The velocity, temperature and concentration profiles are in excellent agreement with the results by Hossain [156]. This implies that the MD-BSLLM and MD-BSQLM methods can be used as numerical methods for solving large parameter non-similar, nonlinear partial differential equations.

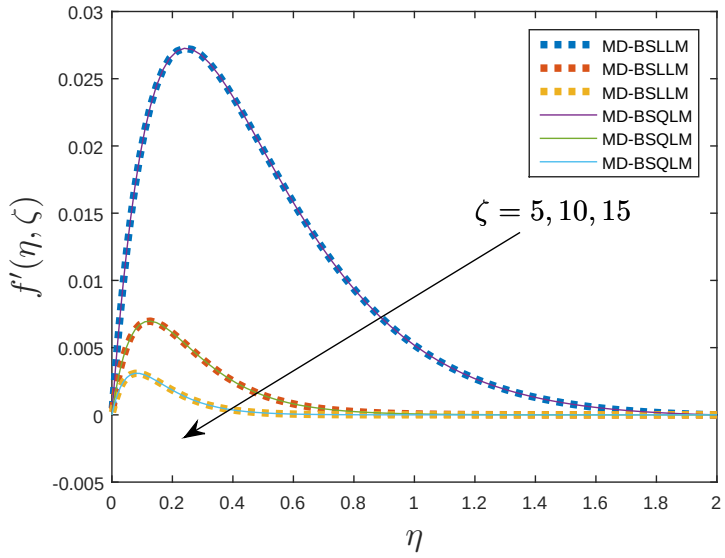

Fig. 8.3 Velocity profiles at different values of ζ, $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$.

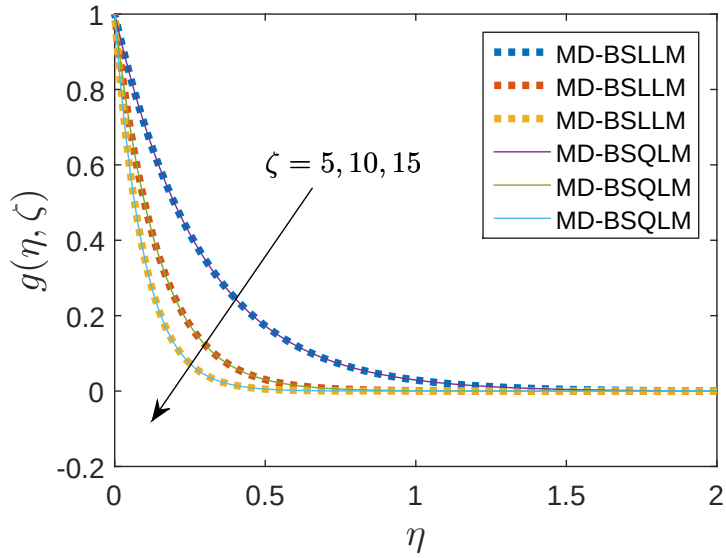

Fig. 8.4 Temperature profiles at different values of ζ, $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$.

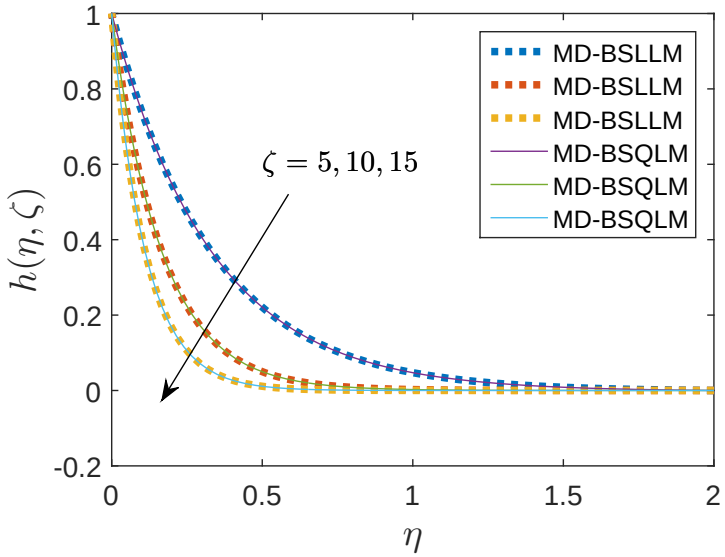

Fig. 8.5 Concentration profiles at different values of ζ, $N_\eta = 60$, $N_\zeta = 5$, $n = w = \frac{1}{2}$, $Pr = 0.7$, $Sc = 0.6$, $q = 40$.

8.7 Conclusion

In this work, new numerical approaches for solving systems of n non-similar nonlinear partial differential equations over large domains are presented and implemented. The methods are generalized for pth order systems of n partial differential equations. The methods were tested by two different numerical experiments. The results were validated by comparison with the series solutions of the numerical experiments. The results of both numerical methods and the series solution were all in excellent agreement which validates the accuracy of the two numerical methods over large domains. The idea of decomposing the main domain into sub-intervals increases the accuracy of the method. The more sub-intervals we have, the more accurate is the method. However, for small domains, it is recommended to use the BSLLM and BSQLM rather than MD-BSLLM and MD-BSQLM respectively because the BSLLM and BSQLM achieve accurate results within a few seconds as opposed to the longer times needed for the multi-domain approach. The proposed numerical methods performs better than some existing numerical methods for solving a class of non-similar boundary layer equations over large time domains with faster convergence and fewer grid points to achieve accurate results. The proposed method uses minimal computation time and its accuracy does not rapidly deteriorate with an increase in

the time domain. This work has added to literature in showing two novel approaches for solving non-similar nonlinear systems of partial differential equations over large domains.

Chapter 9

Conclusion

9.1 Summary of the main findings

In this thesis, for solving nonlinear evolution differential equations and systems of nonlinear partial differential equations, we introduced several new pseudospectral methods that use spectral collocation independently in space and time. These techniques were presented in general form and used to solve a variety of evolution equations, and systems of differential equations that describe physical phenomena. The specific techniques introduced are the bivariate spectral quasilinearization method (BSQLM), bivariate spectral relaxation method (BSRM), bivariate spectral local linearization method (BSLLM), Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (LGL-BSQLM), multi-domain Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (MD-LGL-BSQLM), multi-domain bivariate spectral quasilinearization method (MD-BSQLM), and multi-domain bivariate spectral local linearization method (MD-BSLLM). The use of these methods has been demonstrated, and their accuracy validated.

In Chapter 2, the bivariate spectral quasilinearization for nonlinear evolution equations was introduced. The method was presented in a general form for pth order nonlinear evolution equations. The accuracy and reliability of the BSQLM was confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation.

In Chapter 3, the bivariate spectral quasilinearization method was then applied to a system of nonlinear partial differential equations. The objective was to solve a system of non-similar

boundary layer equations that model magnetohydrodyanamic forced convection flow adjacent to a non-isothermal wedge. It was shown that the method is more accurate than some traditional numerical methods, computationally efficient and robust.

In Chapter 4, the bivariate spectral relaxation method for systems of nonlinear partial differential equations was presented. The spectral method was applied in both space and time. The method was tested on a system of four partial differential equations modeling an unsteady three dimensional magnetohydrodynamic flow and mass transfer in a porous media. Results were compared with previously published results of the SRM, SQLM and the Keller-box method. They showed that the new approach was computationally efficient and converged faster than the other methods, requiring fewer grid points. We conclude that the BSRM offers spectral accuracy in both space and time variables. This accuracy was achieved with many fewer grid points than are needed when using finite differences.

In Chapter 5, the general performance of three bivariate pseudospectral methods (BSQLM, BSLLM and BSRM) was analyzed. We presented, for the first time, generalized algorithms for the BSQLM and BSLLM applicable to systems of n nonlinear system of partial differential equations. We also compared results from the three methods. From the tables showing computational time, we showed that the BSLLM is computationally faster than either of the BSQLM or BSRM.

In Chapter 6, the Legendre-Gauss-Lobbatto bivariate spectral quasilinearization (LGL-BSQLM) for nonlinear evolution equations was introduced. The method was presented in a general form for pth order nonlinear evolution equations. The accuracy and reliability of the LGL-BSQLM was confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation and the modified KdV equation. Error bounds were developed for the first time in this Chapter.

In Chapter 7, the multi-domain Legendre-Gauss-Lobbatto bivariate spectral quasilinearization method (MD-LGL-BSQLM) for nonlinear evolution equations was introduced for the first time. The method was presented in a general form suitable for solving pth order nonlinear evolution equations over a large time interval. The applicability, accuracy and reliability of the proposed MD-LGL-BSQLM was confirmed by solving the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, modified KdV-Burgers equation, and modified KdV equation. The results of the MD-LGL-BSQLM were compared with

known exact solutions that had been reported in the literature, and they agreed to a high degree of accuracy.

In Chapter 8, two new multi-domain decomposition numerical methods for finding solutions of systems of nonlinear coupled non-similar boundary layer partial differential equations over a large time interval were presented. The pseudo-spectral methods, termed the multi-domain bivariate spectral quasilinearization method (MD-BSQLM) and multi-domain bivariate spectral local linearization method (MD-BSLLM), were developed for solving systems of n coupled non-similar boundary layer partial differential equations. The domain was divided into smaller non-overlapping sub-intervals on which the Chebyshev spectral collocation method was used to solve the equations. A continuity condition was used to advance the solution across the sub-intervals. The techniques presented in the chapter were easy to develop and yielded accurate results using few discretization points.

9.2 Future work

Here, we have opened up many new ideas, look at the potential. They have been used to solve various nonlinear differential equations. These ideas can be expanded and applied to different models. The methods described above have been mainly used to model differential equations arising from fluid dynamics, and parabolic nonlinear evolution equations. These methods could also be applied to solve differential equations arising from other fields. New algorithms for hyperbolic and elliptic differential equations could be developed and presented in general forms. Furthermore, only one dimensional nonlinear partial differential equations have been considered in this work. New algorithms could be developed for higher order nonlinear partial differential equations.

On another note, the methods developed in this thesis have been used only for non-periodic boundary conditions. These methods need to be modified to be used easily to solve differential equations with periodic boundary conditions.

References

[1] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

[2] W. Zhang, Q. Chang, and B. Jiang, "Explicit exact solitary-wave solutions for compound kdv-type and compound kdv–burgers-type equations with nonlinear terms of any order," *Chaos, Solitons & Fractals*, vol. 13, no. 2, pp. 311–319, 2002.

[3] M. Sari, G. Gürarslan, and A. Zeytinoğlu, "High-order finite difference schemes for numerical solutions of the generalized burgers–huxley equation," *Numerical Methods for Partial Differential Equations*, vol. 27, no. 5, pp. 1313–1326, 2011.

[4] M. Dehghan, J. M. Heris, and A. Saadatmandi, "Application of semi-analytic methods for the fitzhugh–nagumo equation, which models the transmission of nerve impulses," *Mathematical Methods in the Applied Sciences*, vol. 33, no. 11, pp. 1384–1398, 2010.

[5] S. Abbasbandy, "Approximate solution for the nonlinear model of diffusion and reaction in porous catalysts by means of the homotopy analysis method," *Chemical Engineering Journal*, vol. 136, no. 2, pp. 144–150, 2008.

[6] S. Liao, "On the homotopy analysis method for nonlinear problems," *Applied Mathematics and Computation*, vol. 147, no. 2, pp. 499–513, 2004.

[7] S. Liao, "Notes on the homotopy analysis method: some definitions and theorems," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 4, pp. 983–997, 2009.

[8] J.-H. He, "Comparison of homotopy perturbation method and homotopy analysis method," *Applied Mathematics and Computation*, vol. 156, no. 2, pp. 527–539, 2004.

[9] S. Abbasbandy, "The application of homotopy analysis method to nonlinear equations arising in heat transfer," *Physics Letters A*, vol. 360, no. 1, pp. 109–113, 2006.

[10] M. Ayub, A. Rasheed, and T. Hayat, "Exact flow of a third grade fluid past a porous plate using homotopy analysis method," *International Journal of Engineering Science*, vol. 41, no. 18, pp. 2091–2103, 2003.

[11] J.-H. He, "Homotopy perturbation method: a new nonlinear analytical technique," *Applied Mathematics and Computation*, vol. 135, no. 1, pp. 73–79, 2003.

[12] J.-H. He, "Homotopy perturbation method for solving boundary value problems," *Physics Letters A*, vol. 350, no. 1, pp. 87–88, 2006.

[13] J.-H. He, "Application of homotopy perturbation method to nonlinear wave equations," *Chaos, Solitons & Fractals*, vol. 26, no. 3, pp. 695–700, 2005.

[14] D. D. Ganji, "The application of he's homotopy perturbation method to nonlinear equations arising in heat transfer," *Physics letters A*, vol. 355, no. 4, pp. 337–341, 2006.

[15] D. D. Ganji and A. Sadighi, "Application of he's homotopy-perturbation method to nonlinear coupled systems of reaction-diffusion equations," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 7, no. 4, pp. 411–418, 2006.

[16] A. Siddiqui, R. Mahmood, and Q. Ghori, "Thin film flow of a third grade fluid on a moving belt by he's homotopy perturbation method," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 7, no. 1, pp. 7–14, 2006.

[17] J.-H. He, "Variational iteration method–a kind of non-linear analytical technique: some examples," *International Journal of Non-linear Mechanics*, vol. 34, no. 4, pp. 699–708, 1999.

[18] N. Bildik and A. Konuralp, "The use of variational iteration method, differential transform method and adomian decomposition method for solving different types of nonlinear partial differential equations," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 7, no. 1, pp. 65–70, 2006.

[19] S. Momani and S. Abuasad, "Application of he's variational iteration method to helmholtz equation," *Chaos, Solitons & Fractals*, vol. 27, no. 5, pp. 1119–1123, 2006.

[20] A.-M. Wazwaz, "The variational iteration method for solving linear and nonlinear systems of pdes," *Computers & Mathematics with Applications*, vol. 54, no. 7, pp. 895–902, 2007.

[21] D. D. Ganji and A. Sadighi, "Application of homotopy-perturbation and variational iteration methods to nonlinear heat transfer and porous media equations," *Journal of Computational and Applied Mathematics*, vol. 207, no. 1, pp. 24–34, 2007.

[22] H. B. Keller and T. Cebeci, "Accurate numerical methods for boundary-layer flows. ii: Two dimensional turbulent flows," *AIAA Journal*, vol. 10, no. 9, pp. 1193–1199, 1972.

[23] K. A. Yih, "Uniform suction/blowing effect on forced convection about a wedge: uniform heat flux," *Acta Mechanica*, vol. 128, no. 3-4, pp. 173–181, 1998.

[24] K. A. Yih, "Radiation effect on mixed convection over an isothermal cone in porous media," *Heat and Mass Transfer*, vol. 37, no. 1, pp. 53–57, 2001.

[25] R. Nazar, N. Amin, and I. Pop, "Unsteady boundary layer flow due to a stretching surface in a rotating fluid," *Mechanics Research Communications*, vol. 31, no. 1, pp. 121–128, 2004.

[26] R. L. Burden and J. D. Faires, "Numerical analysis. 2001," *Brooks/Cole, USA*, 2001.

[27] G. D. Smith, *Numerical solution of partial differential equations: finite difference methods*. Oxford University Press, 1985.

[28] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb, *Spectral methods for time-dependent problems*, vol. 21. Cambridge University Press, 2007.

[29] J. P. Boyd, *Chebyshev and Fourier spectral methods*. Courier Corporation, 2001.

[30] M. Y. Hussaini and T. A. Zang, "Spectral methods in fluid dynamics," 1986.

[31] J. Shen, "Efficient spectral-galerkin method i. direct solvers of second-and fourth-order equations using legendre polynomials," *SIAM Journal on Scientific Computing*, vol. 15, no. 6, pp. 1489–1505, 1994.

[32] E. H. Doha and A. H. Bhrawy, "An efficient direct solver for multidimensional elliptic robin boundary value problems using a legendre spectral-galerkin method," *Computers & Mathematics with Applications*, vol. 64, no. 4, pp. 558–571, 2012.

[33] K. Ito and S. S. Ravindran, "A reduced basis method for control problems governed by pdes," in *Control and Estimation of Distributed Parameter Systems*, pp. 153–168, Springer, 1998.

[34] J.-P. Berrut and L. N. Trefethen, "Barycentric lagrange interpolation," *SIAM Review*, vol. 46, no. 3, pp. 501–517, 2004.

[35] B. H. Jung, Y. S. Chung, and T. K. Sarkar, "Time-domain efie, mfie, and cfie formulations using laguerre polynomials as temporal basis functions for the analysis of transient scattering from arbitrary shaped conducting structures," *Progress In Electromagnetics Research*, vol. 39, pp. 1–45, 2003.

[36] Y. Kinoshita and Y. Ohta, "Continuous-time system identification using compactly-supported filter kernels generated from laguerre basis functions," in *49th IEEE Conference on Decision and Control (CDC)*, pp. 4461–4466, IEEE, 2010.

[37] A. H. Khater, R. S. Temsah, and M. M. Hassan, "A chebyshev spectral collocation method for solving burgers-type equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 333–350, 2008.

[38] M. Javidi, "Spectral collocation method for the solution of the generalized burger–fisher equation," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 345–352, 2006.

[39] M. Javidi, "A numerical solution of the generalized burgers–huxley equation by spectral collocation method," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 338–344, 2006.

[40] M. Dehghan and F. Fakhar-Izadi, "Pseudospectral methods for nagumo equation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, no. 4, pp. 553–561, 2011.

[41] L. N. Trefethen, *Spectral methods in MATLAB*, vol. 10. Siam, 2000.

[42] K. A. Yih, "Mhd forced convection flow adjacent to a non-isothermal wedge," *International Communications in Heat and Mass Transfer*, vol. 26, no. 6, pp. 819–827, 1999.

[43] G. Adomian, "Stochastic systems, acad," *Press, NY*, 1983.

[44] G. Adomian, "A review of the decomposition method in applied mathematics," *Journal of Mathematical Analysis and Applications*, vol. 135, no. 2, pp. 501–544, 1988.

[45] L. Bougoffa and R. C. Rach, "Solving nonlocal initial-boundary value problems for linear and nonlinear parabolic and hyperbolic partial differential equations by the adomian decomposition method," *Applied Mathematics and Computation*, vol. 225, pp. 50–61, 2013.

[46] S. Liao, *Advances in the Homotopy Analysis Method*. World Scientific, 2013.

[47] J.-H. He, "Application of homotopy perturbation method to nonlinear wave equations," *Chaos, Solitons & Fractals*, vol. 26, no. 3, pp. 695–700, 2005.

[48] S. Abbasbandy, "The application of homotopy analysis method to solve a generalized hirota–satsuma coupled kdv equation," *Physics Letters A*, vol. 361, no. 6, pp. 478–483, 2007.

[49] L. Song and H. Zhang, "Application of homotopy analysis method to fractional kdv–burgers–kuramoto equation," *Physics Letters A*, vol. 367, no. 1, pp. 88–94, 2007.

[50] E. Parkes and B. Duffy, "An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations," *Computer Physics Communications*, vol. 98, no. 3, pp. 288–300, 1996.

[51] B. R. Duffy and E. J. Parkes, "Travelling solitary wave solutions to a seventh-order generalized kdv equation," *Physics Letters A*, vol. 214, no. 5, pp. 271–272, 1996.

[52] Z. Li, "Exact solitary wave solutions of nonlinear evolution equations," 2000.

[53] Ü. Lepik, "Numerical solution of evolution equations by the haar wavelet method," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 695–704, 2007.

[54] İ. Çelik, "Haar wavelet method for solving generalized burgers-huxley equation," *Arab Journal of Mathematical Sciences*, vol. 18, no. 1, pp. 25–37, 2012.

[55] G. Hariharan, K. Kannan, and K. Sharma, "Haar wavelet method for solving fishers equation," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 284–292, 2009.

[56] J.-H. He and X.-H. Wu, "Exp-function method for nonlinear wave equations," *Chaos, Solitons & Fractals*, vol. 30, no. 3, pp. 700–708, 2006.

[57] C. Chun, "Solitons and periodic solutions for the fifth-order kdv equation with the exp-function method," *Physics Letters A*, vol. 372, no. 16, pp. 2760–2766, 2008.

[58] X.-H. B. Wu and J.-H. He, "Exp-function method and its application to nonlinear equations," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 903–910, 2008.

[59] F. W. Wubs and E. D. de Goede, "An explicit-implicit method for a class of time-dependent partial differential equations," *Applied Numerical Mathematics*, vol. 9, no. 2, pp. 157–181, 1992.

[60] E. M. Elbarbary and M. El-Kady, "Chebyshev finite difference approximation for the boundary value problems," *Applied Mathematics and Computation*, vol. 139, no. 2, pp. 513–523, 2003.

[61] A. Vliegenthart, "On finite-difference methods for the korteweg-de vries equation," *Journal of Engineering Mathematics*, vol. 5, no. 2, pp. 137–155, 1971.

[62] J. Argyris and M. Haase, "An engineer's guide to soliton phenomena: Application of the finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 61, no. 1, pp. 71–122, 1987.

[63] G. F. Carey and Y. Shen, "Approximations of the kdv equation by least squares finite elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 93, no. 1, pp. 1–11, 1991.

[64] K. Djidjeli, W. G. Price, P. Temarel, and E. H. Twizell, "A linearized implicit pseudo-spectral method for certain non-linear water wave equations," *Communications in Numerical Methods in Engineering*, vol. 14, no. 10, pp. 977–993, 1998.

[65] D. Olmos and B. D. Shizgal, "A pseudospectral method of solution of fisher's equation," *Journal of Computational and Applied Mathematics*, vol. 193, no. 1, pp. 219–242, 2006.

[66] T. A. Driscoll, "A composite runge–kutta method for the spectral solution of semilinear pdes," *Journal of Computational Physics*, vol. 182, no. 2, pp. 357–367, 2002.

[67] M. T. Darvishi, S. Kheybari, and F. Khani, "A numerical solution of the korteweg-de vries equation by pseudospectral method using darvishis preconditionings," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 98–105, 2006.

[68] M. T. Darvishi, S. Kheybari, and F. Khani, "Spectral collocation method and darvishis preconditionings to solve the generalized burgers–huxley equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 10, pp. 2091–2103, 2008.

[69] B. A. Jacobs and C. Harley, "Two hybrid methods for solving two-dimensional linear time-fractional partial differential equations," in *Abstract and Applied Analysis*, vol. 2014, Hindawi Publishing Corporation, 2014.

[70] E. Tohidi and A. Kılıçman, "An efficient spectral approximation for solving several types of parabolic pdes with nonlocal boundary conditions," *Mathematical Problems in Engineering*, vol. 2014, 2014.

[71] R. E. Bellman and R. E. Kalaba, "Quasilinearization and nonlinear boundary-value problems," 1965.

[72] A. Golbabai and M. Javidi, "A spectral domain decomposition approach for the generalized burgers–fisher equation," *Chaos, Solitons & Fractals*, vol. 39, no. 1, pp. 385–392, 2009.

[73] A.-M. Wazwaz and A. Gorguis, "An analytic study of fisher's equation by using adomian decomposition method," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 609–620, 2004.

[74] H. Li and Y. Guo, "New exact solutions to the fitzhugh–nagumo equation," *Applied Mathematics and Computation*, vol. 180, no. 2, pp. 524–528, 2006.

[75] E.-G. Fan, "Traveling wave solutions for nonlinear equations using symbolic computation," *Computers & Mathematics with Applications*, vol. 43, no. 6, pp. 671–680, 2002.

[76] Y. N. Kyrychko, M. V. Bartuccelli, and K. B. Blyuss, "Persistence of travelling wave solutions of a fourth order diffusion system," *Journal of Computational and Applied Mathematics*, vol. 176, no. 2, pp. 433–443, 2005.

[77] I. Hashim, M. S. M. Noorani, and M. R. S. Al-Hadidi, "Solving the generalized burgers–huxley equation using the adomian decomposition method," *Mathematical and Computer Modelling*, vol. 43, no. 11, pp. 1404–1411, 2006.

[78] X. Wang, Z. Zhu, and Y. Lu, "Solitary wave solutions of the generalised burgers-huxley equation," *Journal of Physics A: Mathematical and General*, vol. 23, no. 3, p. 271, 1990.

[79] M. A. Helal and M. S. Mehanna, "A comparison between two different methods for solving kdv–burgers equation," *Chaos, Solitons & Fractals*, vol. 28, no. 2, pp. 320–326, 2006.

[80] J. D. Cole, "On a quasilinear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, no. 3, pp. 225–236, 1951.

[81] T. Cebeci and P. Bradshaw, *Physical and computational aspects of convective heat transfer*. Springer Science & Business Media, 2012.

[82] O. Aydın and A. Kaya, "Mhd mixed convection of a viscous dissipating fluid about a permeable vertical flat plate," *Applied Mathematical Modelling*, vol. 33, no. 11, pp. 4086–4096, 2009.

[83] K. V. Prasad, P. Datti, and K. Vajravelu, "Mhd mixed convection flow over a permeable non-isothermal wedge," *Journal of King Saud University-Science*, vol. 25, no. 4, pp. 313–324, 2013.

[84] A. A. Afify and M. A. A. Bazid, "Mhd falkner-skan flow and heat transfer characteristics of nanofluids over a wedge with heat source/sink effects," *Journal of Computational and Theoretical Nanoscience*, vol. 11, no. 8, pp. 1844–1852, 2014.

[85] R. H. Pletcher, J. C. Tannehill, and D. Anderson, *Computational fluid mechanics and heat transfer*. CRC Press, 2012.

[86] T. Watanabe, "Magnetohydrodynamic stability of boundary layers along a flat plate with pressure gradient," *Acta Mechanica*, vol. 65, pp. 41–50, 1987.

[87] T. Watanabe and I. Pop, "Thermal boundary layers in magnetohydrodynamic flow over a flat plate in the presence of a transverse magnetic field," *Acta Mechanica*, vol. 105, no. 1-4, pp. 233–238, 1994.

[88] T. Watanabe and I. Pop, "Hall effects on magnetohydrodynamic boundary layer flow over a continuous moving flat plate," *Acta Mechanica*, vol. 108, no. 1-4, pp. 35–47, 1995.

[89] P. D. Ariel, "Hiemenz flow in hydromagnetics," *Acta Mechanica*, vol. 103, no. 1-4, pp. 31–43, 1994.

[90] A. J. Chamkha, M. Mujtaba, A. Quadri, and C. Issa, "Thermal radiation effects on mhd forced convection flow adjacent to a non-isothermal wedge in the presence of a heat source or sink," *Heat and Mass Transfer*, vol. 39, no. 4, pp. 305–312, 2003.

[91] A. J. Chamkha and A. M. Rashad, "Mhd forced convection flow of a nanofluid adjacent to a non-isothermal wedge," *Computational Thermal Sciences: An International Journal*, vol. 6, no. 1, 2014.

[92] D. Pal and H. Mondal, "Influence of temperature-dependent viscosity and thermal radiation on mhd forced convection over a non-isothermal wedge," *Applied Mathematics and Computation*, vol. 212, no. 1, pp. 194–208, 2009.

[93] D. Pal and H. Mondal, "Influence of thermophoresis and soret–dufour on magnetohydrodynamic heat and mass transfer over a non-isothermal wedge with thermal radiation and ohmic dissipation," *Journal of Magnetism and Magnetic Materials*, vol. 331, pp. 250–255, 2013.

[94] S. Mukhopadhyay and I. C. Mandal, "Boundary layer flow and heat transfer of a casson fluid past a symmetric porous wedge with surface heat flux," *Chinese Physics B*, vol. 23, no. 4, p. 044702, 2014.

[95] Z. Uddin, M. Kumar, and S. Harmand, "Influence of thermal radiation and heat generation/absorption on mhd heat transfer flow of a micropolar fluid past a wedge with hall and ion-slip currents," *Thermal Science*, vol. 18, no. 2, 2014.

[96] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "Spectral relaxation method and spectral quasilinearization method for solving unsteady boundary layer flow problems," *Advances in Mathematical Physics*, vol. 2014, 2014.

[97] S. S. Motsa, V. M. Magagula, and P. Sibanda, "A bivariate chebyshev spectral collocation quasilinearization method for nonlinear evolution parabolic equations," *The Scientific World Journal*, vol. 2014, 2014.

[98] M. Y. Hussaini and T. A. Zang, *Spectral Methods in Fluid Dynamics*. Springer-Verl New York etc., 1988.

[99] T. Hayat, M. Qasim, and Z. Abbas, "Homotopy solution for the unsteady three-dimensional mhd flow and mass transfer in a porous space," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2375–2387, 2010.

[100] S. Liao, *Homotopy Analysis Method in nonlinear differential equations*. Springer, 2012.

[101] Z. Abbas, T. Hayat, M. Sajid, and S. Asghar, "Unsteady flow of a second grade fluid film over an unsteady stretching sheet," *Mathematical and Computer Modelling*, vol. 48, no. 3, pp. 518–526, 2008.

[102] S. M. Ahmad, I., T. Hayat, and M. Ayub, "Unsteady axisymmetric flow of a second-grade fluid over a radially stretching sheet," *Computers & Mathematics with Applications*, vol. 56, no. 5, pp. 1351–1357, 2008.

[103] A. Ali and A. Mehmood, "Homotopy analysis of unsteady boundary layer flow adjacent to permeable stretching surface in a porous medium," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 2, pp. 340–349, 2008.

[104] A. Mehmood, A. Ali, and T. Shah, "Heat transfer analysis of unsteady boundary layer flow by homotopy analysis method," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 5, pp. 902–912, 2008.

[105] A. Alizadeh-Pahlavan and K. Sadeghy, "On the use of homotopy analysis method for solving unsteady mhd flow of maxwellian fluids above impulsively stretching sheets," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 4, pp. 1355–1365, 2009.

[106] T. Fan, H. Xu, and I. Pop, "Unsteady stagnation flow and heat transfer towards a shrinking sheet," *International Communications in Heat and Mass Transfer*, vol. 37, no. 10, pp. 1440–1446, 2010.

[107] H. Xu, S.-J. Liao, and I. Pop, "Series solutions of unsteady three-dimensional mhd flow and heat transfer in the boundary layer over an impulsively stretching plate," *European Journal of Mechanics-B/Fluids*, vol. 26, no. 1, pp. 15–27, 2007.

[108] X. You, H. Xu, and I. Pop, "Homotopy analysis of unsteady heat transfer started impulsively from rest along a symmetric wedge," *International Communications in Heat and Mass Transfer*, vol. 37, no. 1, pp. 47–51, 2010.

[109] A. Ali, N. Amin, and I. Pop, "Unsteady mixed convection boundary layer from a circular cylinder in a micropolar fluid," *International Journal of Chemical Engineering*, vol. 2010, 2010.

[110] A. Fadzilah Md, R. NAZAR, and N. M. ARIFIN, "Numerical solutions of unsteady boundary layer flow due to an impulsively stretching surface," *Journal of Applied Computer Science & Mathematics*, no. 8, p. 4, 2010.

[111] Y. Y. Lok, P. Phang, N. Amin, and I. Pop, "Unsteady boundary layer flow of a micropolar fluid near the forward stagnation point of a plane surface," *International Journal of Engineering Science*, vol. 41, no. 2, pp. 173–186, 2003.

[112] R. Nazar, N. Amin, and I. Pop, "Unsteady boundary layer flow due to a stretching surface in a rotating fluid," *Mechanics Research Communications*, vol. 31, no. 1, pp. 121–128, 2004.

[113] R. Nazar, N. Amin, D. Filip, and I. Pop, "Unsteady boundary layer flow in the region of the stagnation point on a stretching sheet," *International Journal of Engineering Science*, vol. 42, no. 11, pp. 1241–1253, 2004.

[114] S. S. Motsa, F. G. Awad, Z. G. Makukula, and P. Sibanda, "The spectral homotopy analysis method extended to systems of partial differential equations," in *Abstract and Applied Analysis*, vol. 2014, Hindawi Publishing Corporation, 2014.

[115] M. Zayernouri and G. E. Karniadakis, "Exponentially accurate spectral and spectral element methods for fractional odes," *Journal of Computational Physics*, vol. 257, pp. 460–480, 2014.

[116] E. Babolian and M. M. Hosseini, "A modified spectral method for numerical solution of ordinary differential equations with non-analytic solution," *Applied Mathematics and Computation*, vol. 132, no. 2, pp. 341–351, 2002.

[117] J. A. Rad, K. Parand, and S. Kazem, "A numerical investigation to viscous flow over nonlinearly stretching sheet with chemical reaction, heat transfer and magnetic field," *International Journal of Applied and Computational Mathematics*, pp. 1–17, 2016.

[118] D. Breda, O. Diekmann, M. Gyllenberg, F. Scarabel, and R. Vermiglio, "Pseudospectral discretization of nonlinear delay equations: New prospects for numerical bifurcation analysis," *SIAM Journal on Applied Dynamical Systems*, vol. 15, no. 1, pp. 1–23, 2016.

[119] S. Hugues and A. Randriamampianina, "An improved projection scheme applied to pseudospectral methods for the incompressible navier–stokes equations," *International Journal for Numerical Methods in Fluids*, vol. 28, no. 3, pp. 501–521, 1998.

[120] H. C. Ku, T. D. Taylor, and R. S. Hirsh, "Pseudospectral methods for solution of the incompressible navier-stokes equations," *Computers & Fluids*, vol. 15, no. 2, pp. 195–214, 1987.

[121] J. J. Dongarra, B. Straughan, and D. W. Walker, "Chebyshev tau-qz algorithm methods for calculating spectra of hydrodynamic stability problems," *Applied Numerical Mathematics*, vol. 22, no. 4, pp. 399–434, 1996.

[122] E. Serre and J. Pulicani, "A three-dimensional pseudospectral method for rotating flows in a cylinder," *Computers & Fluids*, vol. 30, no. 4, pp. 491–519, 2001.

[123] S. S. Motsa, P. Sibanda, J. M. Ngnotchouye, and G. Marewo, "A spectral relaxation approach for unsteady boundary-layer flow and heat transfer of a nanofluid over a permeable stretching/shrinking sheet," *Advances in Mathematical Physics*, vol. 2014, 2014.

[124] P. K. Kameswaran, P. Sibanda, and S. S. Motsa, "A spectral relaxation method for thermal dispersion and radiation effects in a nanofluid flow," *Boundary Value Problems*, vol. 2013, no. 1, p. 1, 2013.

[125] S. Shateyi and J. Prakash, "A new numerical approach for mhd laminar boundary layer flow and heat transfer of nanofluids over a moving surface in the presence of thermal radiation," *Boundary Value Problems*, vol. 2014, no. 1, p. 1, 2014.

[126] I. S. Oyelakin, S. Mondal, and P. Sibanda, "Unsteady casson nanofluid flow over a stretching sheet with thermal radiation, convective and slip boundary conditions," *Alexandria Engineering Journal*, 2016.

[127] N. A. H. Haroun, P. Sibanda, S. Mondal, and S. S. Motsa, "On unsteady mhd mixed convection in a nanofluid due to a stretching/shrinking surface with suction/injection using the spectral relaxation method," *Boundary Value Problems*, vol. 2015, no. 1, p. 1, 2015.

[128] A. I. Fagbade and A. J. Omowaye, "Influence of thermal radiation on free convective heat and mass transfer past an isothermal vertical oscillating porous plate in the presence of chemical reaction and heat generation-absorption," *Boundary Value Problems*, vol. 2016, no. 1, pp. 1–18, 2016.

[129] N. A. H. Haroun, S. Mondal, and P. Sibanda, "Unsteady natural convective boundary-layer flow of mhd nanofluid over a stretching surfaces with chemical reaction using the spectral relaxation method: A revised model," *Procedia Engineering*, vol. 127, pp. 18–24, 2015.

[130] S. Shateyi, S. S. Motsa, and Z. G. Makukula, "On spectral relaxation method for entropy generation on a mhd flow and heat transfer of a maxwell fluid.," *Journal of Applied Fluid Mechanics*, vol. 8, no. 1, 2015.

[131] S. S. Motsa, Z. G. Makukula, and S. Shateyi, "Numerical investigation of the effect of unsteadiness on three-dimensional flow of an oldroyb-b fluid," *PloS One*, vol. 10, no. 7, p. e0133507, 2015.

[132] V. M. Magagula, S. S. Motsa, P. Sibanda, and P. G. Dlamini, "On a bivariate spectral relaxation method for unsteady magneto-hydrodynamic flow in porous media," *SpringerPlus*, vol. 5, no. 1, p. 1, 2016.

[133] S. S. Motsa and I. L. Animasaun, "A new numerical investigation of some thermo-physical properties on unsteady mhd non-darcian flow past an impulsively started vertical surface," *Thermal Science*, vol. 19, pp. S249–S258, 2015.

[134] S. S. Motsa, "On the new bivariate local linearisation method for solving coupled partial differential equations in some applications of unsteady fluid flows with heat and mass transfer," 2015.

[135] S. S. Motsa, V. M. Magagula, and P. Sibanda, "A bivariate chebyshev spectral collocation quasilinearization method for nonlinear evolution parabolic equations," *The Scientific World Journal*, vol. 2014, 2014.

[136] Z. Abbas, M. Sheikh, and S. S. Motsa, "Numerical solution of binary chemical reaction on stagnation point flow of casson fluid over a stretching/shrinking sheet with thermal radiation," *Energy*, vol. 95, pp. 12–20, 2016.

[137] S. S. Motsa and M. S. Ansari, "Unsteady boundary layer flow and heat transfer of oldroyd-b nanofluid towards a stretching sheet with variable thermal conductivity," *Thermal Science*, vol. 19, no. 1, pp. s239–s248, 2015.

[138] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: evolution to complex geometries and applications to fluid dynamics*. Springer Science & Business Media, 2007.

[139] S. S. Motsa, Z. G. Makukula, and S. Shateyi, "Spectral local linearisation approach for natural convection boundary layer flow," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[140] H. S. Takhar, A. J. Chamkha, and G. Nath, "Unsteady three-dimensional mhd-boundary-layer flow due to the impulsive motion of a stretching surface," *Acta Mechanica*, vol. 146, no. 1-2, pp. 59–71, 2001.

[141] C. RamReddy, T. Pradeepa, and D. Srinivasacharya, "Numerical solution for mixed convection in a darcy porous medium saturated with a micropolar fluid under convective boundary condition using spectral quasi-linearization method," *Advanced Science, Engineering and Medicine*, vol. 7, no. 3, pp. 234–245, 2015.

[142] D. Srinivasacharya, S. S. Motsa, and O. Surender, "Numerical study of free convection in a doubly stratified non-darcy porous medium using spectral quasilinearization method," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 16, no. 3-4, pp. 173–183, 2015.

[143] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.

[144] M. Shamsi and M. Dehghan, "Recovering a time-dependent coefficient in a parabolic equation from overspecified boundary data using the pseudospectral legendre method," *Numerical Methods for Partial Differential Equations*, vol. 23, no. 1, pp. 196–210, 2007.

[145] B. Wu, D. Wang, E. K. Poh, and G. Xu, "Nonlinear optimization of low-thrust trajectory for satellite formation: Legendre pseudospectral approach," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 4, pp. 1371–1381, 2009.

[146] F. Fahroo and I. M. Ross, "Costate estimation by a legendre pseudospectral method," *Journal of Guidance, Control, and Dynamics*, vol. 24, no. 2, pp. 270–277, 2001.

[147] M. Gasca and T. Sauer, "On the history of multivariate polynomial interpolation," *Journal of Computational and Applied Mathematics*, vol. 122, no. 1, pp. 23–35, 2000.

[148] A. H. Bhrawy and M. A. Zaky, "A method based on the jacobi tau approximation for solving multi-term time–space fractional partial differential equations," *Journal of Computational Physics*, vol. 281, pp. 876–895, 2015.

[149] J. Li, H. Ma, and W. Sun, "Error analysis for solving the korteweg-de vries equation by a legendre pseudo-spectral method," *Numerical Methods for Partial Differential Equations*, vol. 16, no. 6, pp. 513–534, 2000.

[150] H. Ma and W. Sun, "A legendre–petrov–galerkin and chebyshev collocation method for third-order differential equations," *SIAM Journal on Numerical Analysis*, vol. 38, no. 5, pp. 1425–1438, 2000.

[151] P. G. Dlamini, M. Khumalo, and S. S. Motsa, "A note on the multi-stage spectral relaxation method for chaos control and synchronization," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 15, no. 5, pp. 289–298, 2014.

[152] S. Shateyi, S. S. Motsa, and Y. Khan, "A new piecewise spectral homotopy analysisof the michaelis-menten enzymatic reactions model," *Numerical Algorithms*, vol. 66, no. 3, pp. 495–510, 2014.

[153] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "A new multistage spectral relaxation method for solving chaotic initial value systems," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 265–283, 2013.

[154] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "Solving hyperchaotic systems using the spectral relaxation method," in *Abstract and Applied Analysis*, vol. 2012, Hindawi Publishing Corporation, 2012.

[155] S. S. Motsa and P. Sibanda, "A multistage linearisation approach to a four-dimensional hyperchaotic system with cubic nonlinearity," *Nonlinear Dynamics*, vol. 70, no. 1, pp. 651–657, 2012.

[156] M. A. Hossain and S. C. Paul, "Free convection from a vertical permeable circular cone with non-uniform surface temperature," *Acta Mechanica*, vol. 151, no. 1-2, pp. 103–114, 2001.

[157] S. Hussain, M. A. Hossain, and M. Wilson, "Natural convection flow from a vertical permeable flat plate with variable surface temperature and species concentration," *Engineering Computations*, vol. 17, no. 7, pp. 789–812, 2000.

[158] C. D. Pruett and C. L. Streett, "A spectral collocation method for compressible, non-similar boundary layers," *International Journal for Numerical Methods in Fluids*, vol. 13, no. 6, pp. 713–737, 1991.

[159] M. Sajid and T. Hayat, "Non-similar series solution for boundary layer flow of a third-order fluid over a stretching sheet," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1576–1585, 2007.

[160] M. A. Hossain and M. S. Munir, "Natural convection flow of a viscous fluid about a truncated cone with temperature-dependent viscosity and thermal conductivity," *International Journal of Numerical Methods for Heat & Fluid Flow*, vol. 11, no. 6, pp. 494–510, 2001.

[161] S. S. Motsa, "A new spectral local linearization method for nonlinear boundary layer flow problems," *Journal of Applied Mathematics*, vol. 2013, 2013.

[162] S. S. Motsa and S. Shateyi, "Numerical analysis of mixed convection magnetohydrodynamic heat and mass transfer past a stretching surface in a micro-polar fluid-saturated porous medium under the influence of ohmic heating," 2013.

[163] S. Shateyi and G. T. Marewo, "On a new numerical analysis of the hall effect on mhd flow and heat transfer over an unsteady stretching permeable surface in the presence of thermal radiation and heat source/sink," *Boundary Value Problems*, vol. 2014, no. 1, pp. 1–17, 2014.

[164] S. Shateyi and G. T. Marewo, "Numerical analysis of unsteady mhd flow near a stagnation point of a two-dimensional porous body with heat and mass transfer, thermal radiation, and chemical reaction," *Boundary Value Problems*, vol. 2014, no. 1, pp. 1–18, 2014.

[165] S. S. Motsa, T. Hayat, and O. Aldossary, "Mhd flow of upper-convected maxwell fluid over porous stretching sheet using successive taylor series linearization method," *Applied Mathematics and Mechanics*, vol. 33, no. 8, pp. 975–990, 2012.

[166] S. Shateyi and G. T. Marewo, "A new numerical approach of mhd flow with heat and mass transfer for the ucm fluid over a stretching surface in the presence of thermal radiation," *Mathematical Problems in Engineering*, vol. 2013, 2013.

[167] S. S. Motsa, "A new spectral local linearization method for nonlinear boundary layer flow problems," *Journal of Applied Mathematics*, vol. 2013, 2013.

[168] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "Spectral relaxation method and spectral quasilinearization method for solving unsteady boundary layer flow problems," *Advances in Mathematical Physics*, vol. 2014, 2014.

[169] Z. Abbas, M. Sheikh, and S. S. Motsa, "Numerical solution of binary chemical reaction on stagnation point flow of casson fluid over a stretching/shrinking sheet with thermal radiation," *Energy*, vol. 95, pp. 12–20, 2016.

[170] S. Motsa, "On the new bivariate local linearisation method for solving coupled partial differential equations in some applications of unsteady fluid flows with heat and mass transfer," 2015.

[171] S. S. Motsa and I. L. Animasaun, "A new numerical investigation of some thermo-physical properties on unsteady mhd non-darcian flow past an impulsively started vertical surface," *Thermal Science*, vol. 19, pp. S249–S258, 2015.

[172] D. Funaro, "A multidomain spectral approximation of elliptic equations," *Numerical Methods for Partial Differential Equations*, vol. 2, no. 3, pp. 187–205, 1986.

[173] H. P. Pfeiffer, L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, "A multidomain spectral method for solving elliptic equations," *Computer Physics Communications*, vol. 152, no. 3, pp. 253–273, 2003.

[174] D. A. Kopriva, "A spectral multidomain method for the solution of hyperbolic systems," *Applied Numerical Mathematics*, vol. 2, no. 3, pp. 221–241, 1986.

[175] R. Peyret, "The chebyshev multidomain approach to stiff problems in fluid mechanics," *Computer Methods in Applied Mechanics and Engineering*, vol. 80, no. 1, pp. 129–145, 1990.

[176] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "A new multistage spectral relaxation method for solving chaotic initial value systems," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 265–283, 2013.

[177] S. S. Motsa, P. G. Dlamini, and M. Khumalo, "Solving hyperchaotic systems using the spectral relaxation method," in *Abstract and Applied Analysis*, vol. 2012, Hindawi Publishing Corporation, 2012.

[178] P. G. Dlamini, M. Khumalo, and S. S. Motsa, "A note on the multi-stage spectral relaxation method for chaos control and synchronization," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 15, no. 5, pp. 289–298, 2014.

[179] H. Saberi Nik and P. Rebelo, "Multistage spectral relaxation method for solving the hyperchaotic complex systems," *The Scientific World Journal*, vol. 2014, 2014.

Appendix A

BSQLM

This article is about the bivariate Chebyshev spectral collocation quasilinearization method for nonlinear evolution parabolic equation. It has been described in detaial in Chapter 2 in this thesis. Part of the results from Chapter 2 were published in this paper.

Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 581987, 13 pages
http://dx.doi.org/10.1155/2014/581987

Research Article

A Bivariate Chebyshev Spectral Collocation Quasilinearization Method for Nonlinear Evolution Parabolic Equations

S. S. Motsa, V. M. Magagula, and P. Sibanda

School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa

Correspondence should be addressed to S. S. Motsa; sandilemotsa@gmail.com

Received 15 July 2014; Accepted 12 August 2014; Published 27 August 2014

Academic Editor: Hassan Saberi Nik

This paper presents a new method for solving higher order nonlinear evolution partial differential equations (NPDEs). The method combines quasilinearisation, the Chebyshev spectral collocation method, and bivariate Lagrange interpolation. In this paper, we use the method to solve several nonlinear evolution equations, such as the modified KdV-Burgers equation, highly nonlinear modified KdV equation, Fisher's equation, Burgers-Fisher equation, Burgers-Huxley equation, and the Fitzhugh-Nagumo equation. The results are compared with known exact analytical solutions from literature to confirm accuracy, convergence, and effectiveness of the method. There is congruence between the numerical results and the exact solutions to a high order of accuracy. Tables were generated to present the order of accuracy of the method; convergence graphs to verify convergence of the method and error graphs are presented to show the excellent agreement between the results from this study and the known results from literature.

1. Introduction

Nonlinearity exists everywhere and, in general, nature is nonlinear. Nonlinear evolution partial differential equations arise in many fields of science, particularly in physics, engineering, chemistry, finance, and biological systems. They are widely used to describe complex phenomena in various fields of sciences, such as wave propagation phenomena, fluid mechanics, plasma physics, quantum mechanics, nonlinear optics, solid state physics, chemical kinematics, physical chemistry, population dynamics, financial industry, and numerous areas of mathematical modeling. The development of both numerical and analytical methods for solving complicated, highly nonlinear evolution partial differential equations continues to be an area of interest to scientists whose research aim is to enrich deep understanding of such alluring nonlinear problems.

Innumerable number of methods for obtaining analytical and approximate solutions to nonlinear evolution equations have been proposed. Some of the analytical methods that have been used to solve evolution nonlinear partial differential equations include Adomian's decomposition method [1–3], homotopy analysis method [4–7], tanh-function method [8–10], Haar wavelet method [11–13], and Exp-function method [14–16]. Several numerical methods have been used to solve nonlinear evolution partial differential equations. These include the explicit-implicit method [17], Chebyshev finite difference methods [18], finite difference methods [19], finite element methods [20], and pseudospectral methods [21, 22].

Some drawbacks of approximate analytical methods include slow convergence, particularly for large time ($t > 1$). They may also be cumbersome to use as some involve manual integration of approximate series solutions and, hence, it is difficult to find closed solutions sometimes. On the other hand, some numerical methods may not work in some cases, for example, when the required solution has to be found near a singularity. Certain numerical methods, for example, finite differences require many grid points to achieve good accuracy and, hence, require a lot of computer memory and computational time. Conventional first-order finite difference methods may result in monotonic and stable solutions, but they are strongly dissipative causing the solution of the strongly convective partial differential equations to become smeared out and often grossly inaccurate. On the other hand,

higher order difference methods are less dissipative but are prone to numerical instabilities.

Spectral methods have been used successfully in many different fields in sciences and engineering because of their ability to give accurate solutions of differential equations. Khater et al. [23] applied the Chebyshev spectral collocation method to solve Burgers type of equations in space and finite differences to approximate the time derivative. The Chebyshev spectral collocation method has been used together with the fourth-order Runge-Kutta method to solve the nonlinear PDEs in this study. The Chebyshev spectral collocation is first applied to the NPDE and this yields a system of ordinary differential equations, which are solved using the fourth-order Runge-Kutta method. Olmos and Shizgal [24], Javidi [25, 26], Dehghan and Fakhar-Izadi [27], Driscoll [28], and Driscoll [28] solved the Fisher, Burgers-Fisher, Burgers-Huxley, Fitzhugh-Nagumo, and KdV equations, respectively, using a combination of the Chebyshev spectral collocation method and fourth-order Runge-Kutta method. Darvishi et al. [29, 30] solved the KdV and the Burgers-Huxley equations using a combination of the Chebyshev spectral collocation method and Darvishi's preconditioning. Jacobs and Harley [31] and Tohidi and Kilicman [32] used spectral collocation directly for solving linear partial differential equations. Accuracy will be compromised if they implement their approach in solving nonlinear partial differential equations since they use Kronecker multiplication.

Chebyshev spectral methods are defined everywhere in the computational domain. Therefore, it is easy to get an accurate value of the function under consideration at any point of the domain, beside the collocation points. This property is often exploited, in particular to get a significant graphic representation of the solution, making the possible oscillations due to a wrong approximation of the derivative apparent. Spectral collocation methods are easy to implement and are adaptable to various problems, including variable coefficient and nonlinear differential equations. The error associated with the Chebyshev approximation is $\mathcal{O}(1/N^r)$ where N refers to the truncation and r is connected to the number of continuous derivatives of the function. The interest in using Chebyshev spectral methods in solving nonlinear PDEs stems from the fact that these methods require less grid points to achieve accurate results. They are computational and efficient compared to traditional methods like finite difference and finite element methods. Chebyshev spectral collocation method has been used in conjunction with additional methods which may have their own drawbacks. Here, we provide an alternative method that is not dependent on another method to approximate the solution.

The main objective of this work is to introduce a new method that uses Chebyshev spectral collocation, bivariate Lagrange interpolation polynomials together with quasilinearisation techniques. The nonlinear evolution equations are first linearized using the quasilinearisation method. The Chebyshev spectral collocation method with Lagrange interpolation polynomials are applied independently in space and time variables of the linearized evolution partial differential equation. This new method is termed bivariate interpolated spectral quasilinearisation method (BI-SQLM). We present the BI-SQLM algorithm in a general setting, where it can be used to solve any rth order nonlinear evolution equations. The applicability, accuracy, and reliability of the proposed BI-SQLM are confirmed by solving the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Cahn-Hillard equation,the fourth-order KdV equation, Fisher's, Burgers-Fisher, Burger-Huxley, and the Fitzhugh-Nagumo equations. The results of the BI-SQLM are compared against known exact solutions that have been reported in the scientific literature. It is observed that the method achieves high accuracy with relatively fewer spatial grid points. It also converges fast to the exact solution and approximates the solution of the problem in a computationally efficient manner with simulations completed in fractions of a second in all cases. Tables are generated to show the order of accuracy of the method and time taken to compute the solutions. It is observed that, as the number of grid points is increased, the error decreases. Error graphs and graphs showing the excellent agreement of the exact and analytical solutions for all the nonlinear evolution equations are also presented.

The paper is organized as follows. In Section 2, we introduce the BI-SQLM algorithm for a general nonlinear evolution PDE. In Section 3, we describe the application of the BI-SQLM to selected test problems. The numerical simulations and results are presented in Section 4. Finally, we conclude in Section 5.

2. Bivariate Interpolated Spectral Quasilinearization Method (BI-SQLM)

In this section, we introduce the *Bivariate Interpolated Spectral Quasilinearization Method* (BI-SQLM) for finding solutions to nonlinear evolution PDEs. Without loss of generality, we consider nonlinear PDEs of the form

$$\frac{\partial u}{\partial \tau} = H\left(u, \frac{\partial u}{\partial \eta}, \frac{\partial^2 u}{\partial \eta^2}, \ldots, \frac{\partial^n u}{\partial \eta^n}\right), \quad \text{with the physical region } \tau \in [0, T], \quad \eta \in [a, b], \tag{1}$$

where n is the order of differentiation, $u(\eta, \tau)$ is the required solution, and H is a nonlinear operator which contains all the spatial derivatives of u. The given physical region, $\tau \in [0, T]$, is converted to the region $t \in [-1, 1]$ using the linear transformation $\tau = T(t + 1)/2$ and $\eta \in [a, b]$ is converted to the region $x \in [-1, 1]$ using the linear transformation

$$\eta = \frac{1}{2}(b - a)\, x + \frac{1}{2}(b + a). \tag{2}$$

Equation (1) can be expressed as

$$\frac{\partial u}{\partial t} = H\left(u, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial x^2}, \ldots, \frac{\partial^n u}{\partial x^n}\right), \quad t \in [-1, 1], \; x \in [-1, 1]. \tag{3}$$

The solution procedure assumes that the solution can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u(x,t) \approx \sum_{i=0}^{N_x}\sum_{j=0}^{N_t} u\left(x_i,t_j\right) L_i(x)\, L_j(t), \tag{4}$$

which interpolates $u(x,t)$ at selected points in both the x and t directions defined by

$$\{x_i\} = \left\{\cos\left(\frac{\pi i}{N_x}\right)\right\}_{i=0}^{N_x}, \qquad \{t_j\} = \left\{\cos\left(\frac{\pi j}{N_t}\right)\right\}_{j=0}^{N_t}. \tag{5}$$

The choice of the Chebyshev-Gauss-Lobatto grid points (5) ensures that there is a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. The functions $L_i(x)$ are the characteristic Lagrange cardinal polynomials

$$L_i(x) = \prod_{\substack{i=0\\ i\neq k}}^{N_x} \frac{x - x_k}{x_i - x_k}, \tag{6}$$

where

$$L_i(x_k) = \delta_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ 1 & \text{if } i = k. \end{cases} \tag{7}$$

The function $L_j(t)$ is defined in a similar manner. Before linearizing (3), it is convenient to split H into its linear and nonlinear components and rewrite the governing equation in the form

$$F\left[u,u',\ldots,u^{(n)}\right] + G\left[u,u',\ldots,u^{(n)}\right] - \dot{u} = 0, \tag{8}$$

where the dot and primes denote the time and space derivatives, respectively, F is a linear operator, and G is a nonlinear operator. Assuming that the difference $u_{r+1} - u_r$ and all it's space derivative is small, we first approximate the nonlinear operator G using the linear terms of the Taylor series and, hence,

$$\begin{aligned} G\left[u,u',\ldots,u^{(n)}\right] &\approx G\left[u_r,u_r',\ldots,u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \frac{\partial G}{\partial u^{(k)}}\left(u_{r+1}^{(k)} - u_r^{(k)}\right), \end{aligned} \tag{9}$$

where r and $r+1$ denote previous and current iterations, respectively. We remark that this quasilinearization method (QLM) approach is a generalisation of the Newton-Raphson method and was first proposed by Bellman and Kalaba [33] for solving nonlinear boundary value problems.

Equation (9) can be expressed as

$$\begin{aligned} G\left[u,u',\ldots,u^{(n)}\right] &\approx G\left[u_r,u_r',\ldots,u_r^{(n)}\right] \\ &\quad + \sum_{k=0}^{n} \phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] u_{r+1}^{(k)} \\ &\quad - \sum_{k=0}^{n} \phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] u_r^{(k)}, \end{aligned} \tag{10}$$

where

$$\phi_{k,r}\left[u_r,u_r',\ldots,u_r^{(n)}\right] = \frac{\partial G}{\partial u^{(k)}}\left[u_r,u_r',\ldots,u_r^{(n)}\right]. \tag{11}$$

Substituting (10) into (8), we get

$$\begin{aligned} &F\left[u_{r+1},u_{r+1}',\ldots,u_{r+1}^{(n)}\right] + \sum_{k=0}^{n} \phi_{k,r} u_{r+1}^{(k)} - \dot{u}_{r+1} \\ &\quad = R_r\left[u_r,u_r',\ldots,u_r^{(n)}\right], \end{aligned} \tag{12}$$

where

$$R_r\left[u_r,u_r',\ldots,u_r^{(n)}\right] = \sum_{k=0}^{n} \phi_{k,r} u_r^{(k)} - G\left[u_r,u_r',\ldots,u_r^{(n)}\right]. \tag{13}$$

A crucial step in the implementation of the solution procedure is the evaluation of the time derivative at the grid points t_j $(j = 0,1,\ldots,N_t)$ and the spatial derivatives at the grid points x_i $(i = 0,1,\ldots,N_x)$. The values of the time derivatives at the Chebyshev-Gauss-Lobatto points (x_i,t_j) are computed as (for $j = 0,1,2,\ldots,N_t$)

$$\begin{aligned} \left.\frac{\partial u}{\partial t}\right|_{x=x_i,t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u\left(x_p,t_k\right) L_p(x_i) \frac{dL_k\left(t_j\right)}{dt} \\ &= \sum_{k=0}^{N_t} u\left(x_i,t_k\right) d_{jk} = \sum_{k=0}^{N_t} d_{jk}\, u\left(x_i,t_k\right), \end{aligned} \tag{14}$$

where $d_{jk} = dL_k(t_j)/dt$ is the standard first derivative Chebyshev differentiation matrix of size $(N_t+1)\times(N_t+1)$ as defined in [34]. The values of the space derivatives at the Chebyshev-Gauss-Lobatto points (x_i,t_j) $(i = 0,1,2,\ldots,N_x)$ are computed as

$$\begin{aligned} \left.\frac{\partial u}{\partial x}\right|_{x=x_i,t=t_j} &= \sum_{p=0}^{N_x}\sum_{k=0}^{N_t} u\left(x_p,t_k\right) \frac{dL_p(x_i)}{dx} L_k\left(t_j\right) \\ &= \sum_{p=0}^{N_x} u\left(x_p,t_j\right) D_{ip} = \sum_{p=0}^{N_x} D_{ip}\, u\left(x_p,t_j\right), \end{aligned} \tag{15}$$

where $D_{ip} = dL_p(x_i)/dx$ is the standard first derivative Chebyshev differentiation matrix of size $(N_x+1)\times(N_x+1)$. Similarly, for an nth order derivative, we have

$$\left.\frac{\partial^n u}{\partial x^n}\right|_{x=x_i,t=t_j} = \sum_{p=0}^{N_x} D_{ip}^n u\left(x_p,t_j\right) = \mathbf{D}^n \mathbf{U}_j, \qquad i = 0,1,2,\ldots,N_x, \tag{16}$$

where the vector $\mathbf{U}_j$ is defined as

$$\mathbf{U}_j = \left[u_j(x_0), u_j(x_1), \ldots, u_j\left(x_{N_x}\right)\right]^T \tag{17}$$

and the superscript T denotes matrix transpose. Substituting (16) into (12) we get

$$F\left[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \ldots, \mathbf{U}^{(n)}_{r+1,j}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t} d_{jk} \mathbf{U}_{r+1,k} = R_r\left[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \ldots, \mathbf{U}^{(n)}_{r,j}\right] \tag{18}$$

for $j = 0, 1, 2, \ldots, N_t$, where

$$\mathbf{U}^{(n)}_{r+1,j} = \mathbf{D}^n \mathbf{U}_{r+1,j},$$

$$\mathbf{\Phi}_{k,r} = \begin{bmatrix} \phi_{k,r}\left(x_0, t_j\right) & & & \\ & \phi_{k,r}\left(x_1, t_j\right) & & \\ & & \ddots & \\ & & & \phi_{k,r}\left(x_{N_x}, t_j\right) \end{bmatrix}. \tag{19}$$

The initial condition for (3) corresponds to $\tau_{N_t} = -1$ and, hence, we express (18) as

$$F\left[\mathbf{U}_{r+1,j}, \mathbf{U}'_{r+1,j}, \ldots, \mathbf{U}^{(n)}_{r+1,j}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{U}^{(k)}_{r+1,j} - \sum_{k=0}^{N_t-1} d_{jk} \mathbf{U}_{r+1,k} = \mathbf{R}_j, \tag{20}$$

where

$$\mathbf{R}_j = R_r\left[\mathbf{U}_{r,j}, \mathbf{U}'_{r,j}, \ldots, \mathbf{U}^{(n)}_{r,j}\right] + d_{jN_t} \mathbf{U}_{N_t}, \quad j = 0, 1, 2, \ldots, N_t - 1. \tag{21}$$

Equation (20) can be expressed as the following $N_t(N_x + 1) \times N_t(N_x + 1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \tag{22}$$

where

$$A_{i,i} = F\left[\mathbf{I}, \mathbf{D}, \ldots, \mathbf{D}^{(n)}\right] + \sum_{k=0}^{n} \mathbf{\Phi}_{k,r} \mathbf{D}^{(k)} - d_{i,i}\mathbf{I}, \quad A_{i,j} = -d_{i,j}\mathbf{I}, \quad \text{when } i \neq j, \tag{23}$$

and $\mathbf{I}$ is the identity matrix of size $(N_x + 1) \times (N_x + 1)$. Solving (19) gives $u(x_i, t_j)$ and, hence, we use (4) to approximate $u(x, t)$.

3. Numerical Experiments

We apply the proposed algorithm to well-known nonlinear PDEs of the form (3) with exact solutions. In order to determine the level of accuracy of the BI-SQLM approximate solution, at a particular time level, in comparison with the exact solution, we report maximum error which is defined by

$$E_N = \max_r \left\{\left|u\left(x_r, t\right) - \widetilde{u}\left(x_r, t\right)\right|, : 0 \leq r \leq N\right\}, \tag{24}$$

where $\widetilde{u}(x_r, t)$ is the approximate solution and is the $u(x_r, t)$ exact solution at the time level t.

Example 1. We consider the generalized Burgers-Fisher equation [35]:

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} \frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u\left(1 - u^{\delta}\right), \tag{25}$$

with initial condition

$$u(x, 0) = \left\{\frac{1}{2} + \frac{1}{2} \tanh\left(\frac{-\alpha\delta}{2(\delta + 1)} x\right)\right\}^{1/\delta} \tag{26}$$

and exact solution

$$u(x, t) = \left\{\frac{1}{2} + \frac{1}{2} \tanh\left(\frac{-\alpha\delta}{2(\delta + 1)} \times \left[x - \left(\frac{\alpha}{\delta + 1} + \frac{\beta(\delta + 1)}{\alpha}\right) t\right]\right)\right\}^{1/\delta}, \tag{27}$$

where α, β, and δ are parameters. For illustration purposes, these parameters are chosen to be $\alpha = \beta = \delta = 1$ in this paper. The linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' + u, \qquad G(u) = -uu' - u^2. \tag{28}$$

We first linearize the nonlinear operator G. We approximate G using the equation

$$G \approx G\left[u_r, u'_r, u''_r\right] + \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_{r+1} - \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_r. \tag{29}$$

The coefficients are given by

$$\phi_{0,r} = \frac{\partial G}{\partial u}\left[u_r, u'_r, u''_r\right] = -\left(u'_r + 2u_r\right), \quad \phi_{1,r} = \frac{\partial G}{\partial u'}\left[u_r, u'_r, u''_r\right] = -u_r, \quad \phi_{2,r} = \frac{\partial G}{\partial u''}\left[u_r, u'_r, u''_r\right] = 0,$$

$$R_r = \sum_{k=0}^{2} \phi_{k,r} u^{(k)}_r - G\left[u_r, u'_r, u''_r\right] = -u_r^2 - u_r u'_r. \tag{30}$$

Therefore, the linearized equation can be expressed as

$$u''_{r+1} + \phi_{1,r} u'_{r+1} + \phi_{0,r} u_{r+1} + u_{r+1} - \dot{u} = R_r. \tag{31}$$

Applying the spectral method both in x and t and initial condition, we get

$$\begin{aligned} &\mathbf{D}^2 \mathbf{U}_{r+1,i} + \mathbf{\Phi}_{1,r} \mathbf{D} \mathbf{U}_{r+1,i} + \mathbf{\Phi}_{0,r} \mathbf{U}_{r+1,i} \\ &\quad + \mathbf{U}_{r+1,i} - 2 \sum_{j=0}^{N_t - 1} d_{ij} \mathbf{U}_{r+1,j} = \mathbf{R}_i. \end{aligned} \tag{32}$$

Equation (32) can be expressed as

$$\begin{aligned} &\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \\ \vdots \\ \mathbf{U}_{N_t-1} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{R}_1 \\ \vdots \\ \mathbf{R}_{N_t-1} \end{bmatrix}, \end{aligned} \tag{33}$$

where

$$\begin{aligned} A_{i,i} &= \mathbf{D}^2 + \mathbf{\Phi}_{1,r}^{(i)} \mathbf{D} + \mathbf{\Phi}_{0,r}^{(i)} + \left(1 - 2d_{i,i}\right) \mathbf{I}, \\ A_{i,j} &= -2d_{i,j} \mathbf{I}, \quad \text{when } i \neq j, \\ \mathbf{R}_i &= R_r + 2d_{iN_t} \mathbf{U}_{r,N_t}. \end{aligned} \tag{34}$$

The boundary conditions are implemented in the first and last row of the matrices A_{ij} and the column vectors $\mathbf{R}_i$ for $i = 0, 1, \ldots, N_t - 1$ and $j = 0, 1, \ldots, N_t - 1$. The procedure for finding the variable coefficients ϕ_i and matrices for the remaining examples is similar.

Example 2. We consider Fisher's equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u (1 - u), \tag{35}$$

subject to the initial condition

$$u(x, 0) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6} x}\right)^2} \tag{36}$$

and exact solution [36]

$$u(x, t) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6} x - 5\alpha t/6}\right)^2}, \tag{37}$$

where α is a constant. The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. For this example, the appropriate linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' + \alpha u, \qquad G(u) = -\alpha u^2. \tag{38}$$

Table 1: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	$1.986e-008$	$1.119e-011$	$7.398e-013$	$7.171e-013$
0.2	$3.934e-008$	$3.121e-011$	$1.552e-012$	$1.561e-012$
0.3	$5.577e-008$	$4.864e-011$	$1.004e-012$	$1.005e-012$
0.4	$6.997e-008$	$6.802e-011$	$7.895e-013$	$8.124e-013$
0.5	$8.107e-008$	$7.971e-011$	$1.088e-012$	$1.027e-012$
0.6	$8.891e-008$	$8.560e-011$	$8.805e-013$	$7.847e-013$
0.7	$9.344e-008$	$8.953e-011$	$6.418e-013$	$6.463e-013$
0.8	$9.431e-008$	$8.759e-011$	$6.199e-013$	$6.164e-013$
0.9	$9.178e-008$	$8.325e-011$	$3.978e-013$	$3.695e-013$
1.0	$8.787e-008$	$7.421e-011$	$7.988e-014$	$5.596e-014$
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

Table 2: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = \gamma = \delta = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	$1.142e-007$	$1.369e-010$	$5.891e-012$	$6.143e-012$
0.2	$1.178e-007$	$1.373e-010$	$9.570e-012$	$1.013e-011$
0.3	$1.186e-007$	$1.479e-010$	$1.489e-011$	$1.512e-011$
0.4	$1.069e-007$	$9.450e-011$	$1.703e-011$	$1.702e-011$
0.5	$9.030e-008$	$7.944e-011$	$5.283e-012$	$5.736e-012$
0.6	$6.963e-008$	$6.618e-011$	$1.639e-011$	$1.626e-011$
0.7	$4.638e-008$	$1.579e-011$	$1.362e-011$	$1.364e-011$
0.8	$2.457e-008$	$4.030e-011$	$3.934e-012$	$3.852e-012$
0.9	$2.028e-008$	$6.006e-011$	$4.466e-012$	$4.727e-012$
1.0	$3.147e-008$	$7.708e-011$	$7.757e-013$	$7.261e-013$
CPU Time (sec)	0.010152	0.015387	0.019163	0.021564

Example 3. Consider the Fitzhugh-Nagumo equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u (u - \alpha) (1 - u) \tag{39}$$

with initial condition

$$u(x, 0) = \frac{1}{2} \left[1 - \coth\left(-\frac{x}{2\sqrt{2}}\right)\right]. \tag{40}$$

This equation has the exact solution [37]

$$u(x, t) = \frac{1}{2} \left[1 - \coth\left(-\frac{x}{2\sqrt{2}} + \frac{2\alpha - 1}{4} t\right)\right], \tag{41}$$

where α is a parameter. In this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \alpha u, \qquad G(u) = (1 + \alpha) u^2 - u^3. \tag{42}$$

Table 3: Maximum errors E_N for the Fitzhug-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	5.719$e-007$	1.196$e-009$	2.367$e-012$	9.881$e-014$
0.2	6.193$e-007$	1.299$e-009$	2.761$e-012$	3.952$e-014$
0.3	6.662$e-007$	1.463$e-009$	3.259$e-012$	8.216$e-014$
0.4	6.779$e-007$	1.448$e-009$	3.341$e-012$	8.094$e-014$
0.5	6.920$e-007$	1.526$e-009$	3.587$e-012$	5.063$e-014$
0.6	7.019$e-007$	1.573$e-009$	3.729$e-012$	3.775$e-014$
0.7	6.933$e-007$	1.516$e-009$	3.660$e-012$	8.915$e-014$
0.8	6.828$e-007$	81.535$e-009$	3.635$e-012$	7.594$e-014$
0.9	6.765$e-007$	1.528$e-009$	3.519$e-012$	3.242$e-013$
1.0	6.687$e-007$	1.490$e-009$	3.405$e-012$	1.688$e-013$
CPU time (sec)	0.024281	0.024901	0.026810	0.032389

Table 4: Maximum errors E_N for the Burger-Huxley equation when $\gamma = 0.75$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	2.217$e-006$	8.482$e-009$	2.166$e-011$	7.822$e-014$
0.2	2.596$e-006$	9.369$e-009$	2.536$e-011$	1.184$e-013$
0.3	2.859$e-006$	1.073$e-008$	3.201$e-011$	1.049$e-013$
0.4	3.001$e-006$	1.112$e-008$	3.652$e-011$	9.426$e-014$
0.5	3.137$e-006$	1.213$e-008$	4.262$e-011$	1.510$e-013$
0.6	3.270$e-006$	1.311$e-008$	4.842$e-011$	2.127$e-013$
0.7	3.367$e-006$	1.359$e-008$	5.289$e-011$	1.230$e-013$
0.8	3.467$e-006$	1.438$e-008$	5.803$e-011$	1.549$e-013$
0.9	3.562$e-006$	1.504$e-008$	6.260$e-011$	3.063$e-013$
1.0	3.640$e-006$	1.559$e-008$	6.674$e-011$	2.951$e-013$
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

Table 5: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	1.803$e-007$	3.419$e-010$	4.449$e-013$	1.572$e-013$
0.2	2.614$e-007$	4.347$e-010$	5.049$e-013$	5.992$e-014$
0.3	2.717$e-007$	4.677$e-010$	5.532$e-013$	8.128$e-013$
0.4	2.009$e-007$	3.663$e-010$	4.771$e-013$	6.158$e-013$
0.5	2.580$e-007$	4.410$e-010$	7.518$e-013$	2.555$e-013$
0.6	2.653$e-007$	4.606$e-010$	8.738$e-013$	5.756$e-013$
0.7	2.248$e-007$	4.039$e-010$	6.210$e-013$	2.393$e-013$
0.8	2.572$e-007$	4.476$e-010$	5.432$e-013$	6.812$e-013$
0.9	2.436$e-007$	4.351$e-010$	6.111$e-013$	6.287$e-013$
1.0	8.275$e-008$	3.721$e-010$	7.569$e-013$	1.087$e-007$
CPU time (sec)	0.015646	0.021226	0.030159	0.035675

Table 6: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.1	7.788$e-005$	3.553$e-007$	7.601$e-010$	2.080$e-010$
0.2	1.153$e-004$	4.000$e-007$	5.684$e-010$	1.189$e-010$
0.3	1.011$e-004$	3.739$e-007$	4.471$e-010$	4.503$e-010$
0.4	3.926$e-005$	1.785$e-007$	6.544$e-010$	4.987$e-010$
0.5	6.727$e-005$	2.342$e-007$	2.638$e-010$	1.528$e-010$
0.6	6.065$e-005$	2.207$e-007$	4.565$e-010$	4.568$e-010$
0.7	2.511$e-005$	1.105$e-007$	4.749$e-010$	3.748$e-010$
0.8	4.074$e-005$	1.427$e-007$	1.062$e-010$	1.604$e-010$
0.9	2.386$e-005$	1.018$e-007$	2.343$e-010$	8.114$e-011$
1.0	1.440$e-006$	7.256$e-008$	1.436$e-009$	1.513$e-011$
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

Example 4. Consider the Burgers-Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} u_x = \frac{\partial^2 u}{\partial x^2} + \beta u \left(1 - u^{\delta}\right)\left(u^{\delta} - \gamma\right), \tag{43}$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study), and $\gamma \in (0, 1)$. The exact solution subject to the initial condition

$$u(x, 0) = \frac{1}{2} - \frac{1}{2} \tanh\left[\frac{\beta}{r - \alpha} x\right], \tag{44}$$

is reported in [38, 39] as

$$u(x, t) = \frac{1}{2} - \frac{1}{2} \tanh\left[\frac{\beta}{r - \alpha}(x - ct)\right], \tag{45}$$

where

$$r = \sqrt{\alpha^2 + 8\beta}, \qquad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4} \tag{46}$$

The general solution (45) was reported in [40, 41]. In this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u'' - \beta\gamma u,$$
$$G(u) = -\alpha u u' + \beta(1 + \gamma) u^2 - \beta u^3. \tag{47}$$

Example 5. We consider the modified KdV-Burgers equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 u}{\partial x^2} - 6u^2 \frac{\partial u}{\partial x} \tag{48}$$

subject to the initial condition

$$u(x, 0) = \frac{1}{6} + \frac{1}{6} \tanh\left(\frac{x}{6}\right) \tag{49}$$

and exact solution [42]

$$u(x, t) = \frac{1}{6} + \frac{1}{6} \tanh\left(\frac{x}{6} - \frac{t}{27}\right). \tag{50}$$

Table 7: Maximum errors E_N for Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$1.119e-011$	$7.398e-013$	$8.266e-013$	$3.808e-014$
0.4	$3.121e-011$	$1.552e-012$	$7.378e-013$	$3.780e-014$
0.6	$4.864e-011$	$1.004e-012$	$3.402e-012$	$7.283e-014$
0.8	$6.802e-011$	$7.895e-013$	$1.118e-012$	$3.714e-014$
1.0	$7.971e-011$	$1.088e-012$	$1.473e-012$	$1.691e-013$
1.2	$8.560e-011$	$8.805e-013$	$2.611e-012$	$3.119e-013$
1.4	$8.953e-011$	$6.418e-013$	$6.671e-012$	$1.796e-013$
1.6	$8.759e-011$	$6.199e-013$	$1.118e-011$	$1.097e-013$
1.8	$8.325e-011$	$3.978e-013$	$7.515e-013$	$6.273e-014$
2.0	$7.421e-011$	$7.988e-014$	$3.682e-012$	$2.311e-013$
CPU time (sec)	0.013542	0.022967	0.023792	0.024758

Table 8: Maximum errors E_N for the Burgers-Fisher equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$1.223e-007$	$1.400e-008$	$1.402e-008$	$1.094e-012$
0.4	$1.145e-007$	$1.919e-008$	$1.918e-008$	$3.919e-012$
0.6	$9.192e-008$	$2.082e-008$	$2.085e-008$	$1.953e-012$
0.8	$2.293e-008$	$1.793e-008$	$1.793e-008$	$6.340e-013$
1.0	$2.395e-008$	$1.337e-008$	$1.339e-008$	$2.381e-012$
1.2	$5.778e-008$	$1.954e-008$	$1.930e-008$	$1.005e-011$
1.4	$6.045e-008$	$1.620e-008$	$1.620e-008$	$3.535e-012$
1.6	$5.244e-008$	$7.218e-009$	$7.345e-009$	$5.765e-012$
1.8	$4.395e-008$	$6.828e-009$	$6.784e-009$	$3.983e-012$
2.0	$2.944e-008$	$9.406e-010$	$8.820e-010$	$3.812e-012$
CPU time (sec)	0.019942	0.025988	0.027756	0.029436

The modified KdV-Burgers equation describes various kinds of phenomena such as a mathematical model of turbulence [43] and the approximate theory of flow through a shock wave traveling in viscous fluid [44]. For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''' - u'', \qquad G(u) = -6u'u^2. \tag{51}$$

Example 6. We consider the high nonlinear modified KdV equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} + \left(\frac{\partial u}{\partial x}\right)^2 - u^2 \tag{52}$$

subject to the initial condition

$$u(x,0) = \frac{1}{2} + \frac{e^{-x}}{4} \tag{53}$$

Table 9: Maximum errors E_N for the Fitzhugh-Nagumo equation when $\alpha = 1$ using $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$6.326e-007$	$1.311e-009$	$2.886e-012$	$1.131e-012$
0.4	$6.721e-007$	$1.467e-009$	$3.310e-012$	$1.564e-012$
0.6	$7.140e-007$	$1.602e-009$	$3.617e-012$	$1.936e-012$
0.8	$6.730e-007$	$1.496e-009$	$4.707e-012$	$1.196e-012$
1.0	$6.660e-007$	$1.487e-009$	$3.675e-012$	$1.264e-012$
1.2	$6.449e-007$	$1.366e-009$	$1.897e-012$	$1.727e-012$
1.4	$5.690e-007$	$1.083e-009$	$2.972e-012$	$1.200e-012$
1.6	$4.931e-007$	$8.010e-010$	$1.519e-012$	$8.590e-013$
1.8	$3.986e-007$	$4.658e-010$	$1.068e-012$	$6.790e-013$
2.0	$2.904e-007$	$2.968e-010$	$1.592e-012$	$1.770e-013$
CPU time (sec)	0.041048	0.049629	0.055008	0.053863

Table 10: Maximum errors E_N for the Burgers-Huxley equation when $\gamma = 0.5$, $\beta = 1$, and $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$2.866e-006$	$1.119e-008$	$3.670e-011$	$1.150e-012$
0.4	$3.401e-006$	$1.420e-008$	$5.744e-011$	$1.638e-012$
0.6	$3.814e-006$	$1.687e-008$	$7.426e-011$	$1.958e-012$
0.8	$3.915e-006$	$1.729e-008$	$8.171e-011$	$7.002e-013$
1.0	$3.938e-006$	$1.738e-008$	$8.157e-011$	$1.267e-012$
1.2	$3.808e-006$	$1.624e-008$	$7.687e-011$	$1.710e-012$
1.4	$3.456e-006$	$1.527e-008$	$6.965e-011$	$5.109e-013$
1.6	$3.230e-006$	$1.349e-008$	$5.535e-011$	$8.203e-013$
1.8	$2.925e-006$	$1.078e-008$	$3.598e-011$	$8.294e-013$
2.0	$2.497e-006$	$7.505e-009$	$2.265e-011$	$9.726e-014$
CPU time (sec)	0.023822	0.024901	0.02685	0.032806

and exact solution

$$u(x,t) = \frac{1}{t+2} + \frac{e^{-(x+t)}}{(t+2)^2}. \tag{54}$$

For this example, the linear operator F and nonlinear operator G are chosen as

$$F(u) = u''', \qquad G(u) = \left(u'\right)^2 - u^2. \tag{55}$$

4. Results and Discussion

In this section we present the numerical solutions obtained using the BI-SQLM algorithm. The number of collocation points in the space x variable used to generate the results is $N_x = 10$ in all cases. Similarly, the number of collocation points in the time t variable used is $N_t = 10$ in all cases. It was found that sufficient accuracy was achieved using these values in all numerical simulations.

Table 11: Maximum errors E_N for the modified KdV-Burgers equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$2.137e-007$	$3.820e-010$	$4.846e-013$	$9.998e-013$
0.4	$2.480e-007$	$4.267e-010$	$5.596e-013$	$8.775e-013$
0.6	$2.691e-007$	$4.676e-010$	$6.565e-013$	$2.054e-012$
0.8	$2.214e-007$	$3.979e-010$	$8.776e-013$	$1.168e-012$
1.0	$2.538e-007$	$4.463e-010$	$9.650e-013$	$8.410e-013$
1.2	$2.650e-007$	$4.680e-010$	$7.450e-013$	$5.113e-013$
1.4	$2.383e-007$	$4.296e-010$	$7.500e-013$	$1.110e-012$
1.6	$2.568e-007$	$4.572e-010$	$9.704e-013$	$2.837e-013$
1.8	$2.520e-007$	$4.529e-010$	$7.443e-013$	$5.353e-013$
2.0	$2.370e-007$	$4.438e-010$	$2.719e-013$	$8.849e-013$
CPU time (sec)	0.062066	0.081646	0.080718	0.10775

Table 12: Maximum errors E_N for the highly nonlinear modified KdV equation, with $N_t = 10$.

$t \setminus N_x$	4	6	8	10
0.2	$1.986e-008$	$1.119e-011$	$7.398e-013$	$7.171e-013$
0.4	$8.010e-005$	$3.577e-007$	$3.902e-008$	$1.979e-010$
0.6	$7.235e-005$	$2.549e-007$	$2.016e-008$	$4.899e-010$
0.8	$6.284e-005$	$1.663e-007$	$1.155e-007$	$2.679e-010$
1.0	$1.642e-005$	$1.620e-007$	$1.243e-007$	$2.474e-010$
1.2	$2.753e-005$	$1.073e-007$	$1.073e-007$	$1.679e-010$
1.4	$3.738e-006$	$8.971e-008$	$8.598e-008$	$4.788e-011$
1.6	$1.223e-005$	$2.153e-008$	$2.503e-008$	$2.941e-011$
1.8	$5.836e-006$	$2.986e-008$	$9.127e-009$	$5.177e-011$
2.0	$9.310e-006$	$6.548e-008$	$7.277e-008$	$1.453e-009$
CPU time (sec)	0.020609	0.021241	0.030617	0.032816

In Tables 1, 2, 3, 4, 5, and 6 we give the maximum errors between the exact and BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at $t \in [0.1, 1]$. The results were computed in the space domain $x \in [0, 1]$. To give a sense of the computational efficiency of the method, the computational time to generate the results is also given. Tables 1–6 clearly show the accuracy of the method. The accuracy is seen to improve with an increase in the number of collocation points N_x. It is remarkable to note that accurate results with errors of order up to 10^{-14} are obtained using very few collocation points in both the x and t variables $N_t \leq 10$, $N_x \leq 10$. This is a clear indication that the BI-SQLM is powerful method that is appropriate in solving nonlinear evolution PDEs. We remark, also, that the BI-SQLM is computationally fast as accurate results are generated in a fraction of a second in all the examples considered in this work.

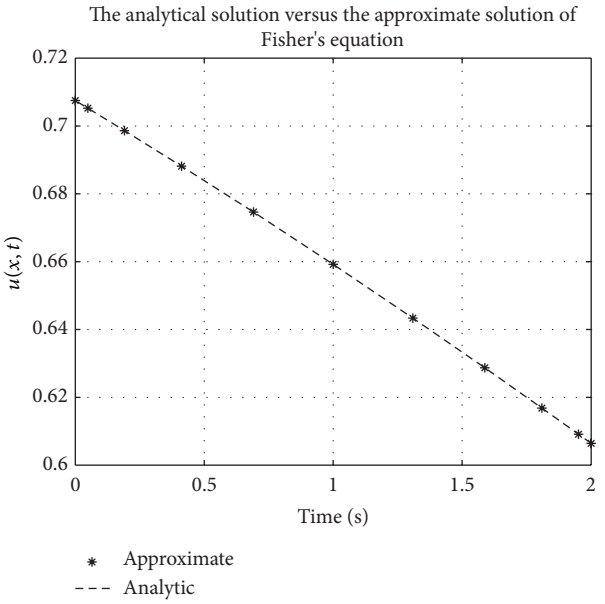

Figure 1: Fishers equation analytical solution graph.

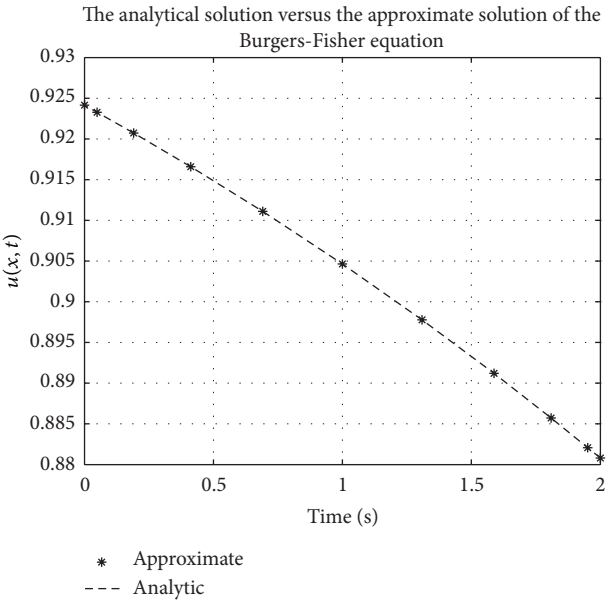

Figure 2: Burger-Fishers equation analytical solution graph.

In Tables 7, 8, 9, 10, 11, and 12 we give the maximum errors of the BI-SQLM results for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, at selected values of $t = 2$ for different collocation points, N_t, in the t-variable. The results in Tables 7–12 were computed on the space domain $x \in [0, 1]$. We note that the accuracy does not detoriate when $t > 1$ for this method as is often the case with numerical schemes such as finite differences.

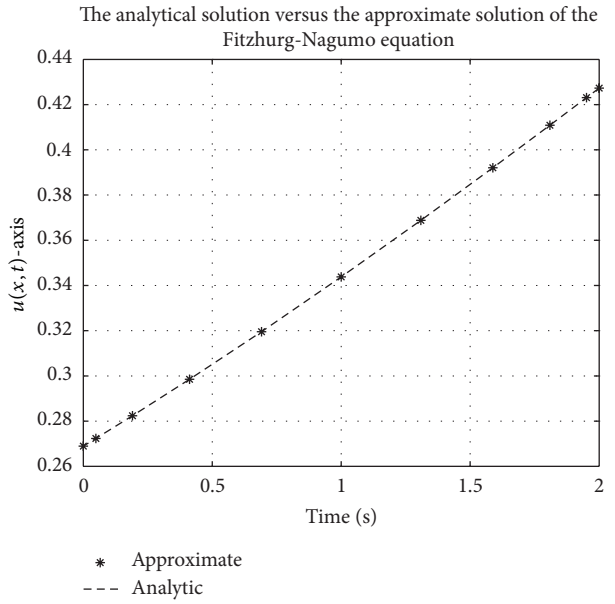

Figure 3: Fitzhugh-Nagumo equation analytical solution graph.

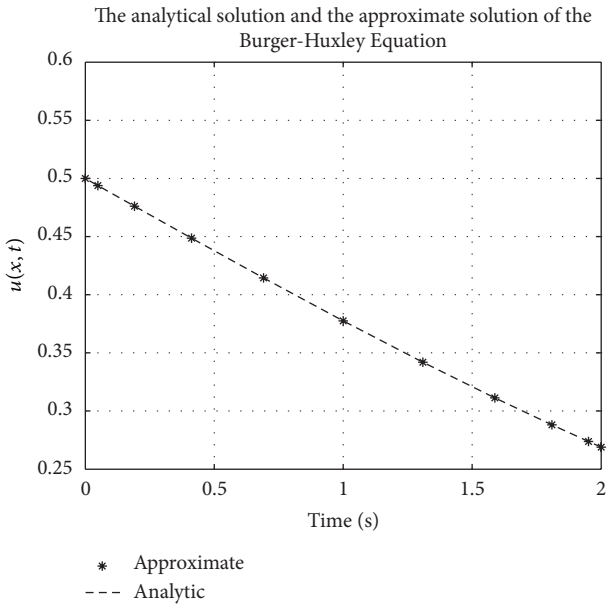

Figure 4: Burgers-Huxley equation analytical solution graph.

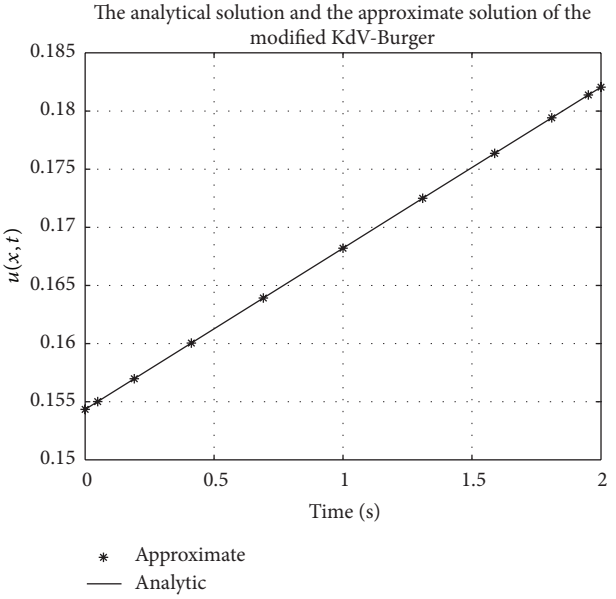

Figure 5: Modified KdV-Burger equation analytical solution graph.

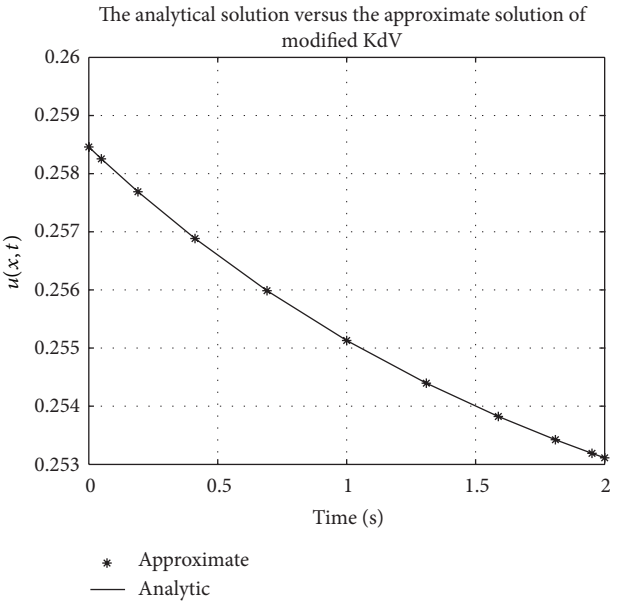

Figure 6: Modified KdV equation analytical solution graph.

Figures 1, 2, 3, 4, 5, and 6 show a comparison of the analytical and approximate solutions of the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$. The approximate solutions are in excellent agreement with the analytical solutions, and this demonstrates the accuracy of the algorithm presented in this study.

In Figures 7, 8, 9, 10, 11, and 12, we present error analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively, when $t = 2$.

In Figures 13, 14, 15, 16, 17, and 18, convergence analysis graphs for the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, Burgers-Huxley equation, the modified KdV-Burgers equation, and the modified KdV equation, respectively. The figures present a variation of the error norm at a fixed value of time ($t = 1$) with iterations of the BI-SQLM scheme. It can be seen that, in almost all the examples considered, the iteration scheme takes about 3 or 4 iterations to converge fully. Beyond the point where full convergence is reached, error norm levels off and does

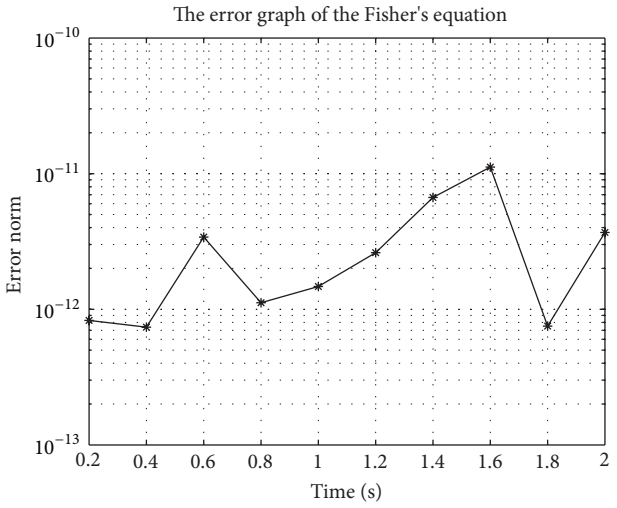

Figure 7: Fishers equation error graph.

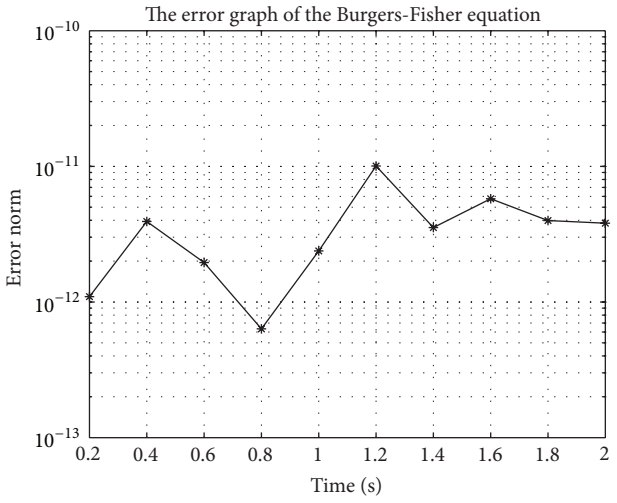

Figure 8: Burger-Fishers equation error graph.

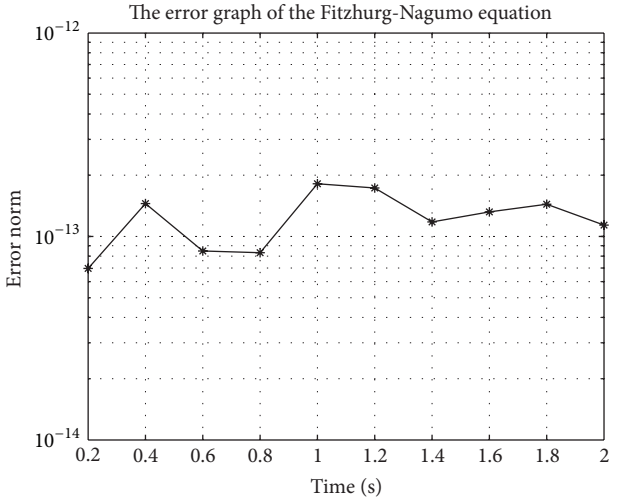

Figure 9: Fitzhugh-Nagumo equation error graph.

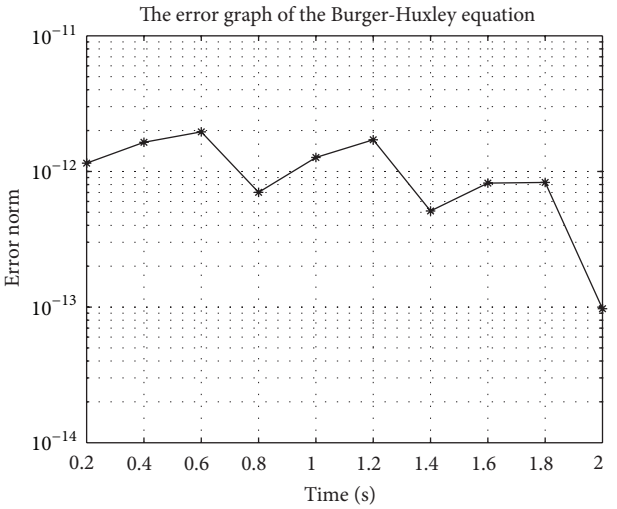

Figure 10: Burgers-Huxley equation error graph.

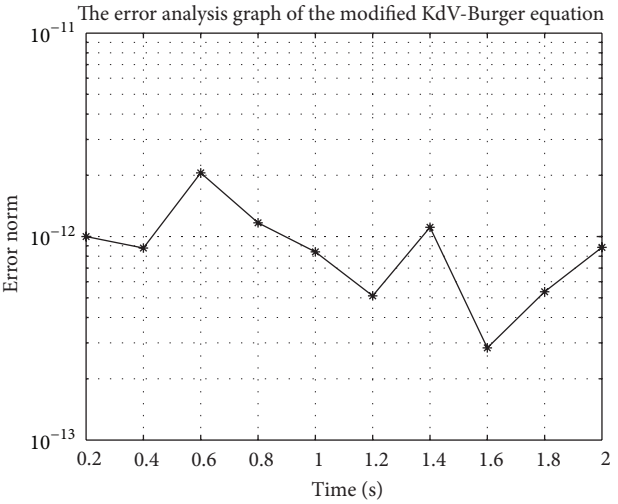

Figure 11: Modified KdV-Burger equation error graph.

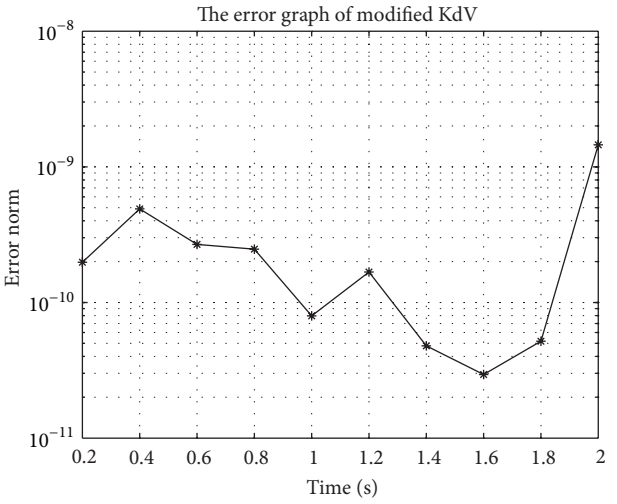

Figure 12: Modified KdV equation error graph.

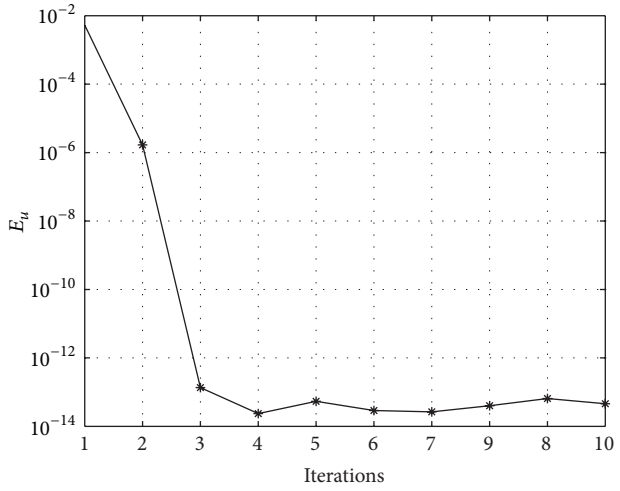

Figure 13: Fishers equation convergence graph.

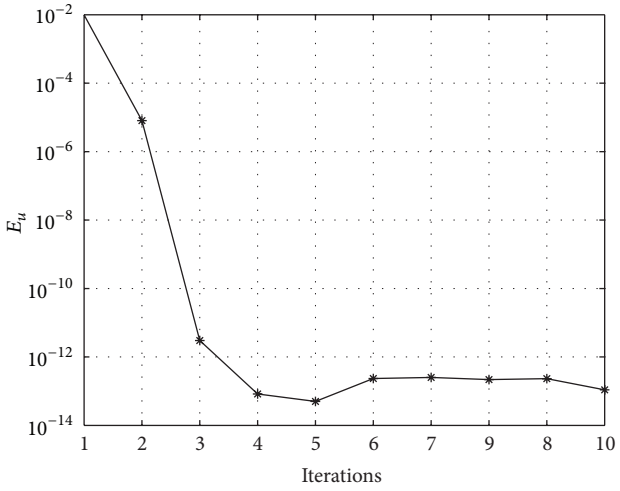

Figure 14: Burger-Fishers equation convergence graph.

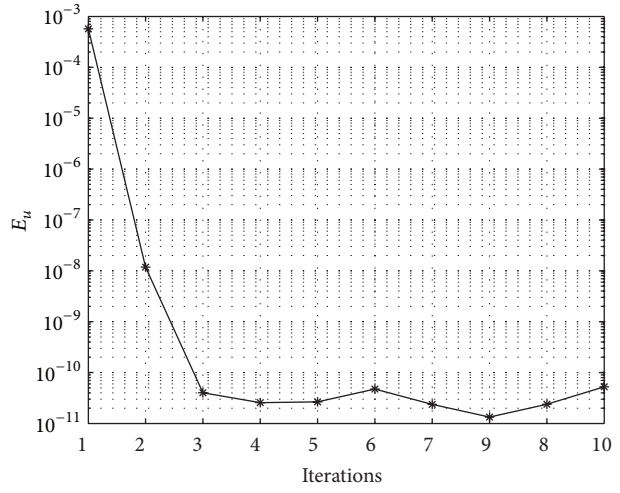

Figure 15: Fitzhugh-Nagumo equation convergence graph.

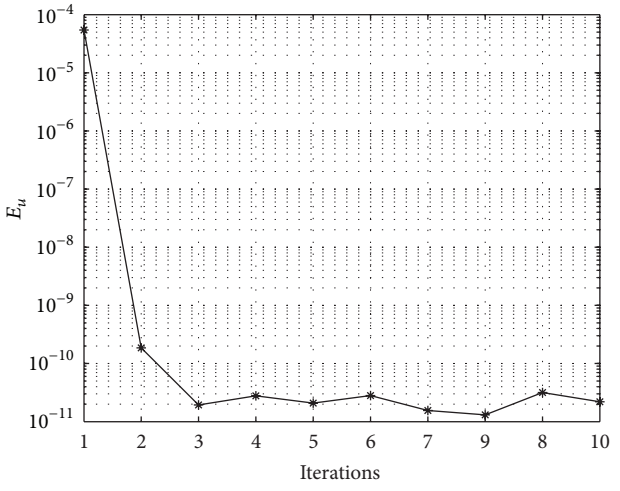

Figure 16: Burgers-Huxley equation convergence graph.

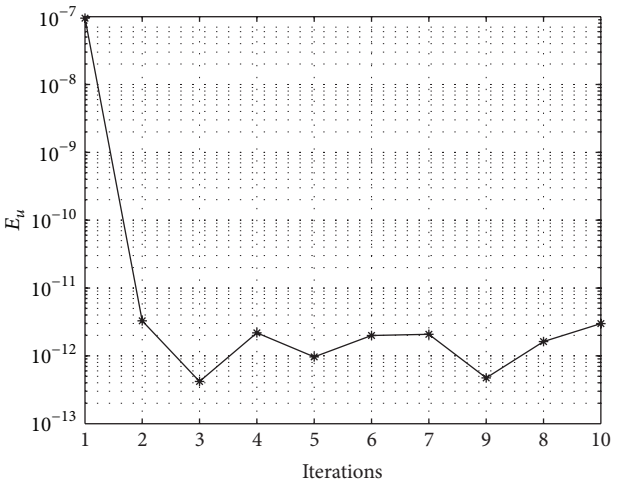

Figure 17: Modified KdV-Burger equation convergence graph.

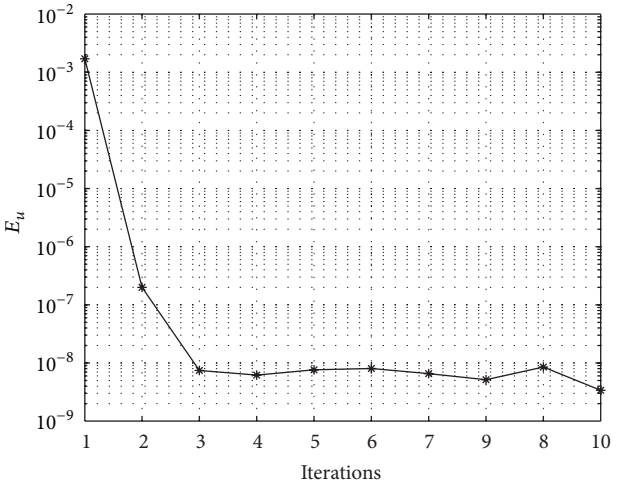

Figure 18: Modified KdV equation convergence graph.

not improve with an increase in the number of iterations. This plateau level gives an estimate of the maximum error that can be achieved when using the proposed method with a certain number of collocation points. It is worth remarking that the accuracy of the method depends on the number of collocation points in both the x and t directions. The results from Figures 13–18 clearly demonstrate that the BI-SQLM is accurate.

5. Conclusion

This paper has presented a new Chebyshev collocation spectral method for solving general nonlinear evolution partial differential equations. The bivariate interpolated spectral quasilinearisation method (BI-SQLM) was developed by combining elements of the quasilinearisation method and Chebyshev spectral collocation with bivariate Lagrange interpolation. The main goal of the current study was to assess the accuracy, robustness, and effectiveness of the method in solving nonlinear partial differential equations.

Numerical simulations were conducted on the modified KdV-Burger equation, highly nonlinear modified KdV equation, the Fisher equation, Burgers-Fisher equation, Fitzhugh-Nagumo equation, and Burgers-Huxley equation. It is evident from the study that the BI-SQLM gives accurate results in a computationally efficient manner. Further evidence from this study is that the BI-SQLM gives solutions that are uniformly accurate and valid in large intervals of space and time domains. The apparent success of the method can be attributed to the use of the Chebyshev spectral collocation method with bivariate Lagrange interpolation in space and time for differentiating. This work contributes to the existing body of literature on quasilinearisation tools for solving complex nonlinear partial differential equations. Further work needs to be done to establish whether the BI-SQLM can be equally successful in solving coupled systems of equations.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported in part by the National Research Foundation of South Africa (Grant no. 85596).

References

[1] G. Adomian, *Stochastic Systems*, vol. 169 of *Mathematics in Science and Engineering*, Academic Press, Orlando, Fla, USA, 1983.

[2] G. Adomian, "A review of the decomposition method in applied mathematics," *Journal of Mathematical Analysis and Applications*, vol. 135, no. 2, pp. 501–544, 1988.

[3] L. Bougoffa and R. C. Rach, "Solving nonlocal initial-boundary value problems for linear and nonlinear parabolic and hyperbolic partial differential equations by the Adomian decomposition method," *Applied Mathematics and Computation*, vol. 225, pp. 50–61, 2013.

[4] S. J. Liao, *Advances in Homotopy Analysis Method*, World Scientific Publishing, Singapore, 2014.

[5] J. He, "Application of homotopy perturbation method to nonlinear wave equations," *Chaos, Solitons and Fractals*, vol. 26, no. 3, pp. 695–700, 2005.

[6] S. Abbasbandy, "The application of homotopy analysis method to solve a generalized Hirota-Satsuma coupled KdV equation," *Physics Letters A: General, Atomic and Solid State Physics*, vol. 361, no. 6, pp. 478–483, 2007.

[7] L. Song and H. Zhang, "Application of homotopy analysis method to fractional KdV-Burgers-KURamoto equation," *Physics Letters A*, vol. 367, no. 1-2, pp. 88–94, 2007.

[8] E. J. Parkes and B. R. Duffy, "An automated tanh-function method for finding solitary wave solutions to non-linear evolution equations," *Computer Physics Communications*, vol. 98, no. 3, pp. 288–300, 1996.

[9] B. R. Duffy and E. J. Parkes, "Travelling solitary wave solutions to a seventh-order generalized KdV equation," *Physics Letters A*, vol. 214, no. 5-6, pp. 271–272, 1996.

[10] Z. B. Li, "Exact solitary wave solutions of nonlinear evolution equations," in *Mathematics Mechanization and Application*, X. S. Gao and D. M. Wang, Eds., Academic Press, San Diego, Calif, USA, 2000.

[11] U. Lepik, "Numerical solution of evolution equations by the Haar wavelet method," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 695–704, 2007.

[12] I. Celik, "Haar wavelet method for solving generalized Burgers-Huxley equation," *Arab Journal of Mathematical Sciences*, vol. 18, no. 1, pp. 25–37, 2012.

[13] G. Hariharan, K. Kannan, and K. R. Sharma, "Haar wavelet method for solving Fisher's equation," *Applied Mathematics and Computation*, vol. 211, no. 2, pp. 284–292, 2009.

[14] J. He and X. Wu, "Exp-function method for nonlinear wave equations," *Chaos, Solitons & Fractals*, vol. 30, no. 3, pp. 700–708, 2006.

[15] C. Chun, "Solitons and periodic solutions for the fifth-order KdV equation with the Exp-function method," *Physics Letters A*, vol. 372, no. 16, pp. 2760–2766, 2008.

[16] X. H. Wu and J. H. He, "EXP-function method and its application to nonlinear equations," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 903–910, 2008.

[17] F. W. Wubs and E. D. de Goede, "An explicit-implicit method for a class of time-dependent partial differential equations," *Applied Numerical Mathematics*, vol. 9, no. 2, pp. 157–181, 1992.

[18] E. M. E. Elbarbary and M. El-Kady, "Chebyshev finite difference approximation for the boundary value problems," *Applied Mathematics and Computation*, vol. 139, no. 2-3, pp. 513–523, 2003.

[19] A. C. Vliegenthart, "On finite-difference methods for the Korteweg-de Vries equation," *Journal of Engineering Mathematics*, vol. 5, pp. 137–155, 1971.

[20] J. Argyris and M. Haase, "An engineer's guide to soliton phenomena: application of the finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 61, no. 1, pp. 71–122, 1987.

[21] G. F. Carey and Y. Shen, "Approximations of the KdV equation by least squares finite elements," *Computer Methods in Applied Mechanics and Engineering*, vol. 93, no. 1, pp. 1–11, 1991.

[22] K. Djidjeli, W. G. Price, P. Temarel, and E. H. Twizell, "A linearized implicit pseudo-spectral method for certain nonlinear water wave equations," *Communications in Numerical*

Methods in Engineering with Biomedical Applications, vol. 14, no. 10, pp. 977–993, 1998.

[23] A. H. Khater, R. S. Temsah, and M. M. Hassan, "A Chebyshev spectral collocation method for solving Burger's-type equations," *Journal of Computational and Applied Mathematics*, vol. 222, no. 2, pp. 333–350, 2008.

[24] D. Olmos and B. D. Shizgal, "A pseudospectral method of solution of Fisher's equation," *Journal of Computational and Applied Mathematics*, vol. 193, no. 1, pp. 219–242, 2006.

[25] M. Javidi, "Spectral collocation method for the solution of the generalized Burger-Fisher equation," *Applied Mathematics and Computation*, vol. 174, no. 1, pp. 345–352, 2006.

[26] M. Javidi, "A numerical solution of the generalized Burgers-Huxley equation by spectral collocation method," *Applied Mathematics and Computation*, vol. 178, no. 2, pp. 338–344, 2006.

[27] M. Dehghan and F. Fakhar-Izadi, "Pseudospectral methods for Nagumo equation," *International Journal for Numerical Methods in Biomedical Engineering*, vol. 27, no. 4, pp. 553–561, 2011.

[28] T. A. Driscoll, "A composite Runge-Kutta method for the spectral solution of semilinear PDEs," *Journal of Computational Physics*, vol. 182, no. 2, pp. 357–367, 2002.

[29] M. T. Darvishi, S. Kheybari, and F. Khani, "A numerical solution of Korteweg-de Vries equation by pseudospectral method using Darvishi's preconditionings," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 98–105, 2006.

[30] M. T. Darvishi, S. Kheybari, and F. Khani, "Spectral collocation method and Darvishi's preconditionings to solve the generalized Burgers-Huxley equation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 10, pp. 2091–2103, 2008.

[31] B. A. Jacobs and C. Harley, "Two hybrid methods for solving two-dimensional linear time-fractional partial differential equations," *Abstract and Applied Analysis*, vol. 2014, Article ID 757204, 10 pages, 2014.

[32] E. Tohidi and A. Kilicman, "An efficient spectral approximation for solving several types of parabolic pdes with nonlocal boundary conditions," *Mathematical Problems in Engineering*, vol. 2014, Article ID 369029, 6 pages, 2014.

[33] R. E. Bellman and R. E. Kalaba, *Quasilinearization and Nonlinear Boundary-Value Problems*, vol. 3 of *Modern Analytic and Computional Methods in Science and Mathematics*, American Elsevier, New York, NY, USA, 1965.

[34] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, Philadelphia, Pa, USA, 2000.

[35] A. Golbabai and M. Javidi, "A spectral domain decomposition approach for the generalized Burger's-Fisher equation," *Chaos, Solitons & Fractals*, vol. 39, no. 1, pp. 385–392, 2009.

[36] A. Wazwaz and A. Gorguis, "An analytic study of Fisher's equation by using Adomian decomposition method," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 609–620, 2004.

[37] H. Li and Y. Guo, "New exact solutions to the FitzHugh-Nagumo equation," *Applied Mathematics and Computation*, vol. 180, no. 2, pp. 524–528, 2006.

[38] E. Fan, "Traveling wave solutions for nonlinear equations using symbolic computation," *Computers & Mathematics with Applications*, vol. 43, no. 6-7, pp. 671–680, 2002.

[39] Y. N. Kyrychko, M. V. Bartuccelli, and K. B. Blyuss, "Persistence of travelling wave solutions of a fourth order diffusion system," *Journal of Computational and Applied Mathematics*, vol. 176, no. 2, pp. 433–443, 2005.

[40] I. Hashim, M. S. M. Noorani, and M. R. Said Al-Hadidi, "Solving the generalized Burgers-Huxley equation using the Adomian decomposition method," *Mathematical and Computer Modelling*, vol. 43, no. 11-12, pp. 1404–1411, 2006.

[41] X. Y. Wang, Z. S. Zhu, and Y. K. Lu, "Solitary wave solutions of the generalised Burgers-Huxley equation," *Journal of Physics A: Mathematical and General*, vol. 23, no. 3, pp. 271–274, 1990.

[42] M. A. Helal and M. S. Mehanna, "A comparison between two different methods for solving KdV-Burgers equation," *Chaos, Solitons and Fractals*, vol. 28, no. 2, pp. 320–326, 2006.

[43] J. M. Burgers, "A mathematical model illustrating the theory of turbulence," in *Advances in Applied Mechanics*, vol. 1, pp. 171–199, 1948.

[44] J. D. Cole, "On a quasi-linear parabolic equation occurring in aerodynamics," *Quarterly of Applied Mathematics*, vol. 9, pp. 225–236, 1951.

Appendix B

BSRM

This article is about the bivariate spectral relaxation method for unsteady three dimensional magneto hydrodynamic flow and mass transfer in a porous media. The material appears in Chapter 4 of this thesis.

Magagula *et al. SpringerPlus (2016) 5:455*
DOI 10.1186/s40064-016-2053-4

SpringerPlus

RESEARCH Open Access

On a bivariate spectral relaxation method for unsteady magneto-hydrodynamic flow in porous media

Vusi M. Magagula[1†], Sandile S. Motsa[1†], Precious Sibanda[1*†] and Phumlani G. Dlamini[2†]

*Correspondence: sibandap@ukzn.ac.za
†All authors contributed equally to this study
[1] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa
Full list of author information is available at the end of the article

Abstract

The paper presents a significant improvement to the implementation of the spectral relaxation method (SRM) for solving nonlinear partial differential equations that arise in the modelling of fluid flow problems. Previously the SRM utilized the spectral method to discretize derivatives in space and finite differences to discretize in time. In this work we seek to improve the performance of the SRM by applying the spectral method to discretize derivatives in both space and time variables. The new approach combines the relaxation scheme of the SRM, bivariate Lagrange interpolation as well as the Chebyshev spectral collocation method. The technique is tested on a system of four nonlinear partial differential equations that model unsteady three-dimensional magneto-hydrodynamic flow and mass transfer in a porous medium. Computed solutions are compared with previously published results obtained using the SRM, the spectral quasilinearization method and the Keller-box method. There is clear evidence that the new approach produces results that as good as, if not better than published results determined using the other methods. The main advantage of the new approach is that it offers better accuracy on coarser grids which significantly improves the computational speed of the method. The technique also leads to faster convergence to the required solution.

Keywords: Bivariate Spectral relaxation method, Magneto-hydrodynamic flow, Porous media

Mathematics Subject Classification: Primary 65N35, 76W99

Introduction

This work describes a new approach to the solution of a system of four partial differential equations that model the flow of unsteady three-dimensional magneto-hydrodynamic flow and mass transfer in porous media. As reported in Hayat et al. (2010), such equations arise in many applications including the aerodynamic extrusion of plastic sheets, the cooling of metallic sheets in a cooling bath and the manufacture of artificial film and fibers. Due to these important applications, many researchers have dedicated time and effort in studying these kind of problems and finding their solutions. The particular model equations considered in this work have been solved in Hayat et al. (2010) using the homotopy analysis method (HAM) and more recently, by Motsa et al. (2014a) using the spectral relaxation method (SRM) and the spectral quasilinearization method

(SQLM). The HAM has been used extensively by researchers working on such problems Abbas et al. (2008), Ahmad et al. (2008), Ali and Mehmood (2008), Mehmood et al. (2008), Alizadeh-Pahlavan and Sadeghy (2009), Fan et al. (2010), Xu et al. (2007), You et al. (2010). It is an analytic method for approximating solutions of differential equations developed by Liao (2012). The homotopy analysis method is an analytic method where accuracy and convergence are achieved by increasing the number of terms of the solution series. In some cases, such as when a large embedded physical parameter multiplies the nonlinear terms, far too many terms may be required to give accurate results. Retaining too many terms in the solution series is cumbersome, even with the use of symbolic computing software. The use of the HAM further depends on other arbitrarily introduced parameters such as the convergence controlling parameter which must be carefully selected through a separate procedure.

A popular numerical method used by many researchers to solve unsteady boundary layer flow problems is the Keller-box method Ali et al. (2010a, b), Lok et al. (2010), Nazar et al. (2004a, b). The Keller-box method is a finite difference based implicit numerical scheme which was developed by Cebeci and Bradshaw (1984). Recently, Motsa et al. (2014a, b) used spectral based relaxation and quasilinearization schemes to solve unsteady boundary layer problems. These schemes are accurate, easy to implement and are computationally efficient. As observed in Motsa et al. (2014a), the limitation of the spectral quasilinearization method is that the coupled high-order system of differential equations may often lead to very large systems of algebraic equations that may require significant computing resources. In addition, the actual process of developing the solution algorithm is time-consuming in comparison to SRM. This is because with SQLM, the process begins with the quasi-linearization step whereas with SRM the iteration scheme is obtained directly by requiring some terms to be evaluated at the current iteration and others at the previous iteration. The SRM works much like the familiar Gauss-Seidel iteration by decoupling a system of non-linear PDEs into a system of linear PDEs which are then solved in succession. Consequently, the SRM is easy to implement and computationally efficient.

Both the original SRM and SQLM used in Motsa et al. (2014a) use finite differences to discretize derivatives in time. This is a disadvantage because finite difference schemes are known to converge slower than spectral methods. The use of finite differences effectively nullifies the benefits of fast convergence when spectral collocation is used to discretize in space. Furthermore, finite differences require fine grids with very small step sizes to guarantee accuracy, hence there is a huge computation time overhead each time the grid is refined. This paper provides a different approach to the implementation of the spectral relaxation method introduced in Motsa et al. (2014a). The innovation is that the spectral collocation method is used to discretize derivatives in both space and time. As a result, there are uniform convergence benefits in both directions. The scheme uses fewer grid points in space and time and thus it converges very fast. We refer to the improved SRM as the bivariate interpolation spectral relaxation method (BI-SRM). To test the viability of this innovation as a solution method, we have solved the coupled system of third and second order partial differential equations that describe a boundary-layer system. A careful comparison of the new results is made with the earlier SRM, SQLM and the

Keller-box results reported in Motsa et al. (2014a). We particularly compare the computational times for the different methods to reach the same level of accuracy.

Model equations

We consider the unsteady and three-dimensional flow of a viscous fluid over a stretching surface investigated by Hayat et al. (2010). The fluid is electrically conducting in the presence of a constant applied magnetic field B_0. The induced magnetic field is neglected under the assumption of a small magnetic Reynolds number. The flow is governed by the following four dimensionless partial differential equations

$$f''' + (1-\xi)\left(\frac{\eta}{2}f'' - \xi\frac{\partial f'}{\partial \xi}\right) + \xi\left[(f+g)f'' - (f')^2 - M^2 f' - \lambda f'\right] = 0, \tag{1}$$

$$g''' + (1-\xi)\left(\frac{\eta}{2}g'' - \xi\frac{\partial g'}{\partial \xi}\right) + \xi\left[(f+g)g'' - (g')^2 - M^2 g' - \lambda g'\right] = 0, \tag{2}$$

$$\begin{aligned} &\theta'' + Pr(1-\xi)\left(\frac{\eta}{2}\theta' - \xi\frac{\partial \theta}{\partial \xi}\right) + Pr\xi(f+g)\theta' = 0, \\ &\phi'' + Sc(1-\xi)\left(\frac{\eta}{2}\phi' - \xi\frac{\partial \phi}{\partial \xi}\right) + Sc\xi(f+g)\phi' - \gamma Sc\xi\phi = 0 \end{aligned} \tag{3}$$

with the following boundary conditions

$$f(\xi,0) = g(\xi,0) = 0, \quad f'(\xi,0) = \theta(\xi,0) = \phi(\xi,0) = 1, \tag{4}$$

$$f'(\xi,\infty) = g'(\xi,\infty) = \theta(\xi,\infty) = \phi(\xi,\infty) = 0, \tag{5}$$

$$g'(\xi,0) = c \tag{6}$$

In the above equations prime denotes the derivative with respect to η, c the stretching parameter is a positive constant. M is the local Hartman number, λ the local porosity parameter, Sc the Schmidt number, Pr the Prandtl number and γ the chemical reaction parameter. The initial unsteady solution can be found exactly by setting $\xi = 0$ in the above equations and solving the resulting equations. The closed form analytical solutions are given by

$$f(0,\eta) = \eta\,\mathrm{erfc}\left(\frac{\eta}{2}\right) + \frac{2}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right], \tag{7}$$

$$g(0,\eta) = c\left(\eta\,\mathrm{erfc}\left(\frac{\eta}{2}\right) + \frac{2}{\sqrt{\pi}}\left[1 - \exp\left(-\frac{\eta^2}{4}\right)\right]\right), \tag{8}$$

$$\theta(0,\eta) = \mathrm{erfc}\left(\frac{\sqrt{Pr}\eta}{2}\right), \tag{9}$$

$$\phi(0,\eta) = \mathrm{erfc}\left(\frac{\sqrt{Sc}\eta}{2}\right). \tag{10}$$

Bivariate interpolated spectral relaxation method (BI-SRM)

In this section we introduce the Bivariate Interpolated Spectral Relaxation Method (BI-SRM) for solving the system of nonlinear partial differential equations (1)–(3). Applying the relaxation scheme Motsa et al. (2014a) to the system of nonlinear partial differential equations gives the following linear partial differential equations;

$$f'''_{r+1} + \alpha_{1,r} f''_{r+1} + \alpha_{2,r} f'_{r+1} + \alpha_{3,r} f_{r+1} - \xi(1-\xi)\frac{\partial f'_{r+1}}{\partial \xi} = R_{1,r}, \tag{11}$$

$$g'''_{r+1} + \beta_{1,r} g''_{r+1} + \beta_{2,r} g'_{r+1} + \beta_{3,r} g_{r+1} - \xi(1-\xi)\frac{\partial g'_{r+1}}{\partial \xi} = R_{2,r}, \tag{12}$$

$$\theta''_{r+1} + \sigma_{1,r}\theta'_{r+1} + \sigma_{2,r}\theta_{r+1} - \mathrm{Pr}\xi(1-\xi)\frac{\partial \theta_{r+1}}{\partial \xi} = R_{3,r}, \tag{13}$$

$$\phi''_{r+1} + \omega_{1,r}\phi'_{r+1} + \omega_{2,r}\phi_{r+1} - \mathrm{Sc}\xi(1-\xi)\frac{\partial \phi_{r+1}}{\partial \xi} = R_{4,r}, \tag{14}$$

subject to

$$\begin{aligned} &f_{r+1}(\xi,0) = g_{r+1}(\xi,0) = 0, \quad f'_{r+1}(\xi,0) = \theta_{r+1}(\xi,0) = \phi_{r+1}(\xi,0) = 1, \\ &f'_{r+1}(\xi,\infty) = g'_{r+1}(\xi,\infty) = \theta_{r+1}(\xi,\infty) = \phi_{r+1}(\xi,\infty) = 0, \\ &g'_{r+1}(\xi,0) = c \end{aligned} \tag{15}$$

where the variable coefficients are given by

$$\begin{aligned} &\alpha_{1,r} = \frac{1}{2}\eta(1-\xi) + \xi g_r, \quad \alpha_{2,r} = -\xi(M^2+\lambda), \quad \alpha_{3,r} = 0, \\ &\beta_{1,r} = \frac{1}{2}\eta(1-\xi) + \xi f_r, \quad \beta_{2,r} = -\xi(M^2+\lambda), \quad \beta_{3,r} = 0, \\ &\sigma_{1,r} = \mathrm{Pr}\left(\frac{1}{2}\eta(1-\xi) + \xi(f_r+g_r)\right), \quad \sigma_{2,r} = 0 \\ &\omega_{1,r} = \mathrm{Sc}\left(\frac{1}{2}\eta(1-\xi) + \xi(f_r+g_r)\right), \quad \omega_{2,r} = -\gamma\,\mathrm{Sc}\,\xi \\ &R_{1,r} = \xi(f')_r^2 - \xi f_r f''_r, \quad R_{2,r} = \xi(g')_r^2 - \xi g_r g''_r, \quad R_{3,r} = 0, \quad R_{4,r} = 0 \end{aligned}$$

and r and $r+1$ denote previous and current iterations respectively. The system of linear partial differential equations (11)–(14) is discretised using the Chebyshev spectral collocation both in space (η) and time (ξ) directions. The Chebyshev collocation method is valid in the domain $[-1,1]$ in space and time. Therefore, the physical region, $\xi \in [0,1]$ is converted to the region $t \in [-1,1]$ using a linear transformation and similarly, $\eta \in [0, L_\infty]$ is converted to the region $x \in [-1,1]$. The system of linear partial differential equations (11)–(14) is decoupled. Therefore, each equation can be solved independently of the other equations in the system. We assume that the solution to Eq. (11) can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$f(\eta,\xi) \approx \sum_{p=0}^{N_x}\sum_{j=0}^{N_t} f(x_p,t_j)L_p(x)L_j(t), \tag{16}$$

which interpolates $f(\eta,\xi)$ at selected points in both the η and ξ directions defined by

$$\{x_p\} = \left\{\cos\left(\frac{\pi p}{N_x}\right)\right\}_{p=0}^{N_x}, \qquad \{t_j\} = \left\{\cos\left(\frac{\pi j}{N_t}\right)\right\}_{j=0}^{N_t}. \tag{17}$$

The Chebyshev–Gauss–Lobatto grid points (17) ensures that there is a simple conversion of the continuous derivatives, in both space and time, to discrete derivatives at the grid points. The characteristic Lagrange cardinal polynomial $L_p(x)$ is defined as

$$L_p(x) = \prod_{\substack{p=0 \\ p\neq s}}^{N_x} \frac{x - x_s}{x_p - x_s}, \tag{18}$$

where

$$L_p(x_s) = \delta_{ps} = \begin{cases} 0 & \text{if } p \neq s \\ 1 & \text{if } p = s \end{cases} \tag{19}$$

Similarly, we define the function $L_j(t)$. Equation (16) is then substituted into Eq. (11). An important step in the implementation of the solution procedure is the evaluation of the derivatives of $L_p(x)$ and $L_j(t)$ with respect to x and t respectively. The derivative of $f(\eta,\xi)$ with respect to ξ at the Chebyshev–Gauss–Lobatto points (x_s, t_i), is computed as

$$\left.\frac{\partial f}{\partial \xi}\right|_{(x_s,t_i)} = 2\sum_{p=0}^{N_x}\sum_{j=0}^{N_t} f(x_p,t_j) L_p(x_s) \frac{dL_j(t_i)}{dt} \tag{20}$$

$$= 2\sum_{j=0}^{N_t} d_{ij} f(x_s,t_j) = \sum_{j=0}^{N_t} d_{ij}\mathbf{F}_j \tag{21}$$

where $d_{ij} = \frac{dL_j(t_i)}{dt}$ are the entries of the standard first derivative Chebyshev differentiation matrix $\mathbf{d} = [d_{ij}]$ of size $(N_t+1)\times(N_t+1)$ as defined in Trefethen (2000) for $i,j = 0,1,\ldots,N_t$. Similarly, we compute the derivative of $f(\eta,\xi)$ with respect to η at the Chebyshev-Gauss-Lobatto points (x_s, t_i), as follows

$$\left.\frac{\partial f}{\partial \eta}\right|_{(x_s,t_i)} = \left(\frac{2}{L_\infty}\right)\sum_{p=0}^{N_x}\sum_{j=0}^{N_t} f(x_p,t_j)\frac{dL_p(x_s)}{dx} L_j(t_i) \tag{22}$$

$$= \left(\frac{2}{L_\infty}\right)\sum_{p=0}^{N_x} D_{sp} f(x_p,t_i) = \mathbf{D}\mathbf{F}_i, \tag{23}$$

where $D_{sp} = \frac{dL_p(x_s)}{dx}$, are the entries of the standard first derivative Chebyshev differentiation matrix of size $(N_x+1)\times(N_x+1)$. Therefore, an nth order derivative with respect to η is given by

$$\left.\frac{\partial^n f}{\partial \eta^n}\right|_{(x_s,t_j)} = \left(\frac{2}{L_\infty}\right)^n \sum_{p=0}^{N_x} D_{sp}^n f(x_p,t_i) = \mathbf{D}^n\mathbf{F}_i, \quad i = 0,1,2,\ldots,N_x, \tag{24}$$

The vector $\mathbf{F}_i$ is defined as

$$\mathbf{F}_i = [f_i(x_0), f_i(x_1), \ldots, f_i(x_{N_x})]^T. \tag{25}$$

where the superscript T denotes matrix transpose. Collocating using Eqs. (24) and (21) on (11), we get

$$\left[\mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r}\right]\mathbf{F}_{r+1,i} - \xi_i(1-\xi_i)\sum_{j=0}^{N_t} d_{ij}\mathbf{D}\mathbf{F}_{r+1,j} = \mathbf{R}_{1,r}, \quad i = 0, 1, 2, \ldots, N_t \tag{26}$$

where $\boldsymbol{\alpha}_{v,r}$ $(v = 1, 2, 3)$ is the diagonal matrix of the vector $[\alpha_{v,r}(x_0), \alpha_{v,r}(x_1), \ldots, \alpha_{v,r}(x_{N_x})]^T$ and $\mathbf{R}_{1,r} = [R_{1,r}(x_0), R_{1,r}(x_1), \ldots, R_{1,r}(x_{N_x})]^T$. The boundary equations are given by

$$f_{r+1,i}(x_{N_x}) = 0, \quad f'_{r+1,i}(x_{N_x}) = 1, \quad f'_{r+1,i}(x_0) = 0, \tag{27}$$

The initial unsteady solution given by equation (7) corresponds to $t = t_{N_t} = -1$. Therefore, we evaluate Eq. (26) for $i = 0, 1, \ldots, N_t - 1$. Equation (26) can be expressed as

$$\left[\mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r}\right]\mathbf{F}_{r+1,i} - \xi_i(1-\xi_i)\sum_{j=0}^{N_t-1} d_{ij}\mathbf{D}\mathbf{F}_{r+1,j} = \mathbf{R}_{1,i}, \quad i = 0, 1, 2, \ldots, N_t, \tag{28}$$

where

$$\mathbf{R}_{1,i} = \mathbf{R}_{1,r} + \xi_i(1-\xi_i)d_{iN_t}\mathbf{D}\mathbf{F}_{N_t},$$

and $\mathbf{F}_{N_t}$ is the known initial unsteady solution given by equation (7). Imposing boundary conditions for $i = 0, 1, \ldots, N_t - 1$, Eq. (28) can be expressed as the following $N_t(N_x + 1) \times N_t(N_x + 1)$ matrix system

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N_t-1,0} & A_{N_t-1,1} & \cdots & A_{N_t-1,N_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{F}_0 \\ \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_{N_t-1} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{1,0} \\ \mathbf{R}_{1,1} \\ \vdots \\ \mathbf{R}_{1,N_t-1} \end{bmatrix}, \tag{29}$$

where

$$A_{i,i} = \mathbf{D}^3 + \boldsymbol{\alpha}_{1,r}\mathbf{D}^2 + \boldsymbol{\alpha}_{2,r}\mathbf{D} + \boldsymbol{\alpha}_{3,r} - \xi_i(1-\xi_i)d_{ii}\mathbf{D} \tag{30}$$

$$A_{i,j} = -\xi_i(1-\xi_i)d_{ij}\mathbf{D}, \quad \text{when } i \neq j, \tag{31}$$

Imposing initial boundary conditions and applying the Chebyshev bivariate collocation as described above on Eqs. (12), (13) and (14) we get

$$\left[\mathbf{D}^3 + \boldsymbol{\beta}_{1,r}\mathbf{D}^2 + \boldsymbol{\beta}_{2,r}\mathbf{D} + \boldsymbol{\beta}_{3,r}\right]\mathbf{G}_{r+1,i} - \xi_i(1-\xi_i)\sum_{j=0}^{N_t-1} d_{ij}\mathbf{D}\mathbf{G}_{r+1,j} = \mathbf{R}_{2,i}, \tag{32}$$

$$\left[\mathbf{D}^2 + \boldsymbol{\sigma}_{1,r}\mathbf{D} + \boldsymbol{\sigma}_{2,r}\right]\boldsymbol{\Theta}_{r+1,i} - \mathrm{Pr}\xi_i(1-\xi_i)\sum_{j=0}^{N_t-1} d_{ij}\boldsymbol{\Theta}_{r+1,j} = \mathbf{R}_{3,i}, \tag{33}$$

$$\left[\mathbf{D}^2+\boldsymbol{\omega}_{1,r}\mathbf{D}+\boldsymbol{\omega}_{2,r}\right]\boldsymbol{\Phi}_{r+1,i}-\mathrm{Sc}\xi_i(1-\xi_i)\sum_{j=0}^{N_t-1}d_{ij}\boldsymbol{\Phi}_{r+1,j}=\mathbf{R}_{4,i}, \tag{34}$$

where

$$\mathbf{R}_{2,i}=\mathbf{R}_{2,r}+\xi_i(1-\xi_i)d_{iN_t}\mathbf{D}\mathbf{G}_{N_t}, \tag{35}$$

$$\mathbf{R}_{3,i}=\mathbf{R}_{3,r}+\mathrm{Pr}\xi_i(1-\xi_i)d_{iN_t}\boldsymbol{\Theta}_{N_t}, \tag{36}$$

$$\mathbf{R}_{4,i}=\mathbf{R}_{4,r}+\mathrm{Sc}\xi_i(1-\xi_i)d_{iN_t}\boldsymbol{\Phi}_{N_t}, \tag{37}$$

and the vectors $\mathbf{G}_{N_t}$, $\boldsymbol{\Theta}_{N_t}$ and $\boldsymbol{\Phi}_{N_t}$ are the known initial unsteady solutions given by Eqs. (8), (9) and (10) respectively. Imposing boundary conditions for $i=0,1,\ldots,N_t-1$, equations (32), (33) and (34) can be expressed as the following $N_t(N_x+1)\times N_t(N_x+1)$ matrix system

$$\begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,N_t-1}\\ B_{1,0} & B_{1,1} & \cdots & B_{1,N_t-1}\\ \vdots & \vdots & \ddots & \vdots\\ B_{N_t-1,0} & B_{N_t-1,1} & \cdots & B_{N_t-1,N_t-1}\end{bmatrix}\begin{bmatrix}\mathbf{G}_0\\ \mathbf{G}_1\\ \vdots\\ \mathbf{G}_{N_t-1}\end{bmatrix}=\begin{bmatrix}\mathbf{R}_{2,0}\\ \mathbf{R}_{2,1}\\ \vdots\\ \mathbf{R}_{2,N_t-1}\end{bmatrix}, \tag{38}$$

$$\begin{bmatrix} C_{0,0} & C_{0,1} & \cdots & C_{0,N_t-1}\\ C_{1,0} & C_{1,1} & \cdots & C_{1,N_t-1}\\ \vdots & \vdots & \ddots & \vdots\\ C_{N_t-1,0} & C_{N_t-1,1} & \cdots & C_{N_t-1,N_t-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Theta}_0\\ \boldsymbol{\Theta}_1\\ \vdots\\ \boldsymbol{\Theta}_{N_t-1}\end{bmatrix}=\begin{bmatrix}\mathbf{R}_{3,0}\\ \mathbf{R}_{3,1}\\ \vdots\\ \mathbf{R}_{3,N_t-1}\end{bmatrix}, \tag{39}$$

$$\begin{bmatrix} E_{0,0} & E_{0,1} & \cdots & E_{0,N_t-1}\\ E_{1,0} & E_{1,1} & \cdots & E_{1,N_t-1}\\ \vdots & \vdots & \ddots & \vdots\\ E_{N_t-1,0} & E_{N_t-1,1} & \cdots & E_{N_t-1,N_t-1}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_0\\ \boldsymbol{\Phi}_1\\ \vdots\\ \boldsymbol{\Phi}_{N_t-1}\end{bmatrix}=\begin{bmatrix}\mathbf{R}_{4,0}\\ \mathbf{R}_{4,1}\\ \vdots\\ \mathbf{R}_{4,N_t-1}\end{bmatrix}, \tag{40}$$

where

$$B_{i,i}=\mathbf{D}^3+\boldsymbol{\beta}_{1,r}\mathbf{D}^2+\boldsymbol{\beta}_{2,r}\mathbf{D}+\boldsymbol{\beta}_{3,r}-\xi_i(1-\xi_i)d_{ii}\mathbf{D} \tag{41}$$

$$B_{i,j}=-\xi_i(1-\xi_i)d_{ij}\mathbf{D},\quad \text{when } i\neq j, \tag{42}$$

$$C_{i,i}=\mathbf{D}^2+\boldsymbol{\sigma}_{1,r}\mathbf{D}+\boldsymbol{\sigma}_{2,r}-\mathrm{Pr}\xi_i(1-\xi_i)d_{ii}\mathbf{I} \tag{43}$$

$$C_{i,j}=-\mathrm{Pr}\xi_i(1-\xi_i)d_{ij}\mathbf{I},\quad \text{when } i\neq j, \tag{44}$$

$$E_{i,i}=\mathbf{D}^2+\boldsymbol{\omega}_{1,r}\mathbf{D}+\boldsymbol{\omega}_{2,r}-\mathrm{Sc}\xi_i(1-\xi_i)d_{ii}\mathbf{I} \tag{45}$$

$$E_{i,j}=-\mathrm{Sc}\xi_i(1-\xi_i)d_{ij}\mathbf{I},\quad \text{when } i\neq j, \tag{46}$$

and $\mathbf{I}$ is the standard $(N_x+1)\times(N_x+1)$ identity matrix. We obtain the numerical solutions for $g(\eta,\xi)$, $\theta(\eta,\xi)$ and $\phi(\eta,\xi)$ by solving matrix equations (38), (39) and (40)

iteratively for $r = 1, 2, \ldots M$, where M is the number of iterations to be used. Equations (8), (9) and (10) are used as initial guesses.

Results and discussion

In this section we present the numerical solutions of the three dimensional unsteady three dimensional magneto-hydrodynamic flow and mass transfer in a porous media obtained using the BI-SQLM algorithm. In our computations the η domain was truncated to $\eta_\infty = 20$. This value gave accurate results for all the quantities of physical interest. To get accurate solutions, $N_x = 60$ collocation points were used to discretize the space variable η and only $N_t = 10$ collocation points were enough for the time variable ξ.

As earlier mentioned, this problem has been solved before by Motsa et al. (2014a) using the spectral relaxation method (SRM), spectral quasilinearization method (SQLM) and the Keller-box method. The results from their paper combined with the present results of the BI-SRM are shown in Table 1. It can be observed from the table that the Keller-box method takes a significant amount of computational time than the SRM and SQLM. This is because the Keller-box is entirely based on finite difference schemes while the SRM and SQLM only uses finite differences in the time variable. In the space variable both the SRM and SQLM use spectral method. It is well documented from literature

Table 1 Values of $f''(0,\xi)$, $g''(0,\xi)$, $\theta'(0,\xi)$ and $\phi'(0,\xi)$ when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

ξ	BI-SRM ($N_t = 10$)	SRM ($N_t = 2000$)	SQLM ($N_t = 2000$)	Keller-box ($N_t = 2000$)
	$f''(0,\xi)$			
0.1	−0.851257	−0.851257	−0.851257	−0.851257
0.3	−1.316705	−1.316705	−1.316705	−1.316705
0.5	−1.685306	−1.685306	−1.685306	−1.685306
0.7	−1.992608	−1.992608	−1.992608	−1.992608
0.9	−2.259335	−2.259335	−2.259335	−2.259335
	$g''(0,\xi)$			
0.1	−0.417150	−0.417150	−0.417150	−0.417150
0.3	−0.639602	−0.639602	−0.639602	−0.639602
0.5	−0.817649	−0.817649	−0.817649	−0.817649
0.7	−0.966603	−0.966603	−0.966603	−0.966603
0.9	−1.095983	−1.095983	−1.095983	−1.095983
	$\theta'(0,\xi)$			
0.1	−0.710882	−0.710882	−0.710882	−0.710882
0.3	−0.742842	−0.742842	−0.742842	−0.742842
0.5	−0.765244	−0.765244	−0.765244	−0.765244
0.7	−0.777270	−0.777270	−0.777270	−0.777270
0.9	−0.770807	−0.770807	−0.770807	−0.770807
	$\phi'(0,\xi)$			
0.1	−0.634443	−0.634443	−0.634443	−0.634443
0.3	−0.766867	−0.766867	−0.766867	−0.766867
0.5	−0.891207	−0.891207	−0.891207	−0.891207
0.7	−1.010045	−1.010045	−1.010045	−1.010045
0.9	−1.125549	−1.125549	−1.125549	−1.125549
CPU time	0.47	18.90	83.24	900.30

Table 2 Values of $f''(0,\xi)$, $g''(0,\xi)$, $\theta'(0,\xi)$ and $\phi'(0,\xi)$ when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

$\xi \backslash N_t$	5	10	15	20
	$f''(0,\xi)$			
0.1	−0.85118289	−0.85125725	−0.85125723	−0.85125724
0.3	−1.31678885	−1.31670509	−1.31670508	−1.31670508
0.5	−1.68525477	−1.68530619	−1.68530619	−1.68530619
0.7	−1.99262557	−1.99260827	−1.99260827	−1.99260827
0.9	−2.25932899	−2.25933501	−2.25933501	−2.25933501
	$g''(0,\xi)$			
0.1	−0.41712280	−0.41715041	−0.41715040	−0.41715040
0.3	−0.63962554	−0.63960199	−0.63960200	−0.63960200
0.5	−0.81764133	−0.81764898	−0.81764898	−0.81764898
0.7	−0.96660357	−0.96660340	−0.96660340	−0.96660340
0.9	−1.09598187	−1.09598304	−1.09598304	−1.09598304
	$\theta'(0,\xi)$			
0.1	−0.71037577	−0.71087263	−0.71088151	−0.71088162
0.3	−0.74357007	−0.74283215	−0.74284211	−0.74284190
0.5	−0.76493472	−0.76524664	−0.76524370	−0.76524351
0.7	−0.77722565	−0.77727704	−0.77727014	−0.77727005
0.9	−0.77086219	−0.77080729	−0.77080662	−0.77080662
	$\phi'(0,\xi)$			
0.1	−0.63444437	−0.63444336	−0.63444326	−0.63444326
0.3	−0.76685596	−0.76686699	−0.76686689	−0.76686689
0.5	−0.89121805	−0.89120664	−0.89120666	−0.89120666
0.7	−1.01004174	−1.01004487	−1.01004494	−1.01004494
0.9	−1.12555031	−1.12554912	−1.12554913	−1.12554913
CPU time	0.13	0.47	1.25	2.47

that spectral methods converge very fast when the solution is smooth. This brought about the idea of using spectral methods in both space and time to increase efficiency. The BI-SRM discretizes both the space and time domains using spectral methods. From the results shown in the table it is evident that the BI-SRM is by far superior than the other methods in terms of computational time taken to reach the same level of accuracy. In Table 1, we also show the number of grid points required by each of the methods to discretize in time. All the finite difference based discretizations required 2000 grid points compared to the spectral discretization of the BI-SRM which required only 10 grid points to reach the same level of accuracy.

The grid independence test for the algorithm is shown in Table 2. The skin friction values, Nusselt number and Sherwood numbers are the variables used in carrying out the grid independence test in Table 2.

The residual error graphs of Eqs. (1)–(3) are presented in Figs. 1, 2, 3 and 4 respectively. In Figs. 1 and 2, we observe that the residual error is reduced with an increase in the iterations of the scheme. The rate of reduction of the residual error appears to be linear. The residual error is minimum at $\xi = 0$ and is increased sharply near 0 until a certain level is reached after which it is almost constant. The residual error appears to be nearly uniform in $0 < \xi \leq 1$ or increases only slightly. It is also observed that the

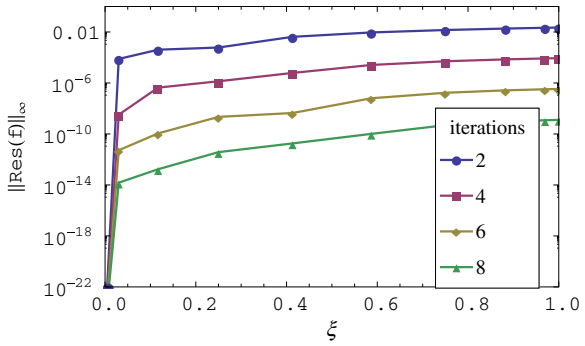

Fig. 1 Residual graph of $f(\eta, \xi)$

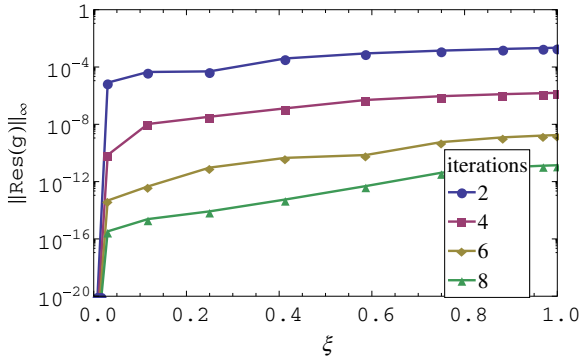

Fig. 2 Residual graph of $g(\eta, \xi)$

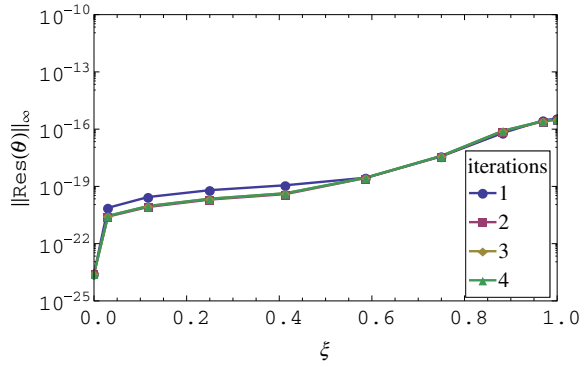

Fig. 3 Residual graph of $\theta(\eta, \xi)$

order of magnitude of the residual error can be seen to be small in the $0 \leq \xi \leq 1$ interval. Lastly, after only two iterations the residual error appears to be less than 0.01 in the entire range of ξ. The small residual error using only a few iteration points to the accuracy of the method. This error can be decreased at a linear rate with an increase in the number of iterations. The decrease in the error with additional iterations suggests that the iteration scheme converges. It should be noted that when $\xi = 0$, governing equations reduce to a linear system that can be solved directly using the spectral collocation

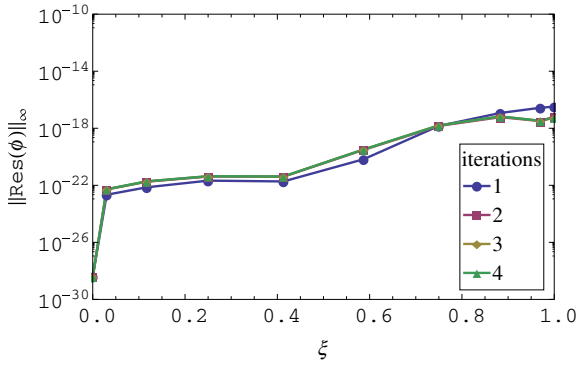

Fig. 4 Residual graph of $\phi(\eta, \xi)$

method with discretization only in η without the use of relaxation and iterations. This explains why the best accuracy is observed at $\xi = 0$. The near uniformity of the residual error in $0 < \xi \leq 1$ can be attributed to the use of Lagrange polynomial basis functions whose error is known to be uniformly distributed in the interpolating region. We can therefore conclude that the method gives accurate results, the rate of convergence of the method is linear and that the method requires only a few iterations to give very accurate results.

We observe that the residual error for the first iteration appears to be very small in Figs. 3 and 4. The residual error is the same for all iterations greater than one. The residual error increases with an increase in ξ. We also observe that the residual error is smaller than the one for f and g even when fewer iterations are used. The observation that the residual error for the momentum and energy is small even for the 1st iteration is perhaps the most interesting finding of the study. This means that when using the proposed approach, the best possible results that can be achieved by the method can be obtained after using just one iteration. Further increase in the number of iterations doesn't improve the accuracy of the solution. After one iteration of the momentum equations for f and g the energy and mass transfer equations reduce to linear homogeneous equations whose solution appears to be marginally influenced by variations in f_r and g_r for $r > 1$. Since with just one iteration we obtain extremely accurate results for θ and ϕ, the implication is that in solving for energy and momentum equations for such a problem, it is not necessary to iterate. It is enough to just use the initial approximation. Is is worth noting that the energy and mass transfer equations are homogeneous equations in θ and ϕ respectively. It is possible that the findings obtained in this study are only applicable in such equations. This has to be investigated further.

The convergence graphs of Eqs. (1)–(3) are presented in Figs. 5, 6 respectively. In Figs. 5 and 6, the residual error decreases linearly with an increase in the number of iterations. The residual error is smallest when ξ is near zero and largest when ξ is large. This is seen from the convergence level which is near 10^{-20} for $\xi = 0.25$ and about 10^{-15} for $\xi = 0.75$. The slope of the residual error graphs is the same for all values of ξ. Full convergence is achieved after 13 iterations for both $\xi = 0.75$ and $\xi = 1$. For $\xi = 0.25$ full convergence is achieved after 16 iterations but at a much smaller magnitude of residual

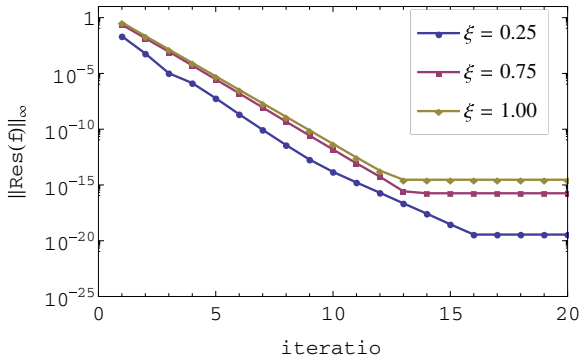

Fig. 5 Convergence graph of $f(\eta, \xi)$

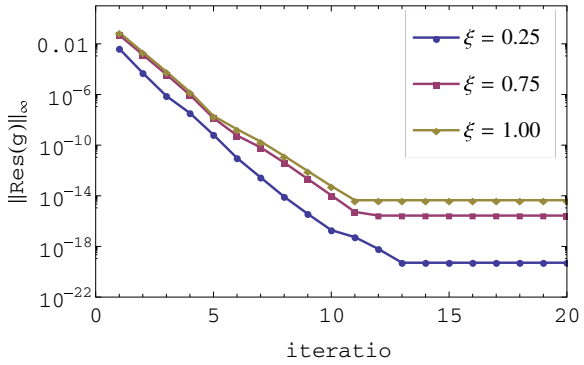

Fig. 6 Convergence graph of $g(\eta, \xi)$

error. The decrease in the residual error with increase in iterations suggests that the iteration scheme converges. Small residual error near zero suggests that best accuracy (after full convergence) is observed near zero. The method converges (fewer iterations needed to attain full convergence) at or near $\xi = 1$. However, the convergence efficiency doesn't translate to better accuracy because, as can be seen for the case of ξ values near zero, the convergence level is 10^{-16}. The same slope for all the graphs means that the convergence rates of the method is the same for all values of ξ. The method is convergent and very accurate in whole time domain $\xi \in [0, 1]$ which translates to $\tau \in [0, \infty)$. The method converges with nearly the same convergence rate for all values of ξ. The method gives the best accuracy near $\xi = 0$ and less accurate, comparatively, at or near $\xi = 1$. We note that even at $\xi = 1$, the method gives very accurate results with a residual error norm of about 10^{-15}. This is one of the highlights of this investigation.

The accuracy of the solutions for energy and mass transfer equations are not dependent on successive relaxation or iterations of the momentum equations since the convergence of the solutions doesn't improve at all with an increase in the number of iterations. Hence, the results for $\theta(\eta)$ are not dependent on successive approximations for $f(\eta)$ and $g(\eta)$. It was observed that it is enough to use one iteration of $f(\eta)$ to give accurate result for $\theta(\eta)$.

In Table 1, we present the results we obtained using the algorithm. The skin friction values, Nusselt number and Sherwood number for various values of ξ are displayed in Table 1. The results obtained using the method match those obtained using other methods. Computing time for the BI-SRM is much smaller than the other methods it is compared with. The method gives valid results when used with few collocation points. In particular, the BI-SRM requires only ten grid points to achieve the same valid results. The table validates the results obtained in this study. BI-SRM is computationally fast in generating valid results when compared with the SRM, SQLM and Keller-Box. We can infer that the BI-SRM is better than finite differences coupled SRM in terms of computational speed and accuracy and the better accuracy could be the result of applying spectral collocation with uniform accuracy level in both η and ξ directions.

In Tables 3 and 4, the residual errors and convergence rates of g and g when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1$ are displayed. We observe that the convergence rate is linear and the residual error is smaller near $\xi = 0$.

Table 3 The residual errors and convergence rates of f when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

Iter.	$\|Res(f)\|_\infty$			Convergence rates		
	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$
1	2.14×10^{-2}	2.36×10^{-1}	3.72×10^{-1}	1.14	1.00	0.98
2	6.03×10^{-4}	1.43×10^{-2}	2.24×10^{-2}	0.50	1.01	1.01
3	1.05×10^{-5}	8.58×10^{-4}	1.43×10^{-3}	1.53	1.00	1.00
4	1.36×10^{-6}	5.04×10^{-5}	8.85×10^{-5}	1.06	1.01	1.00
5	5.95×10^{-8}	2.98×10^{-6}	5.52×10^{-6}	0.97	1.02	1.00
6	2.18×10^{-9}	1.70×10^{-7}	3.40×10^{-7}	0.99	1.00	1.00
7	8.84×10^{-11}	9.06×10^{-9}	2.07×10^{-8}	0.95	1.00	1.00
8	3.75×10^{-12}	4.85×10^{-10}	1.27×10^{-9}	0.85	0.99	1.00

Table 4 The residual errors and convergence rates of g when $\lambda = 0.5, M = 2, c = 0.5, Sc = \gamma = 1, Pr = 1.5$

Iter.	$\|Res(g)\|_\infty$			Convergence Rates		
	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$	$\xi = 0.25$	$\xi = 0.75$	$\xi = 1.00$
1	4.17×10^{-3}	4.93×10^{-2}	7.83×10^{-2}	0.94	1.01	0.99
2	4.96×10^{-5}	1.40×10^{-3}	2.21×10^{-3}	0.75	1.03	1.04
3	7.66×10^{-7}	3.81×10^{-5}	6.39×10^{-5}	1.26	1.13	1.18
4	3.38×10^{-8}	9.21×10^{-7}	1.58×10^{-6}	1.07	0.76	0.56
5	6.56×10^{-10}	1.38×10^{-8}	2.01×10^{-8}	0.83	0.68	0.92
6	9.49×10^{-12}	5.75×10^{-10}	1.78×10^{-9}	1.00	1.27	1.18
7	2.82×10^{-13}	6.58×10^{-11}	1.93×10^{-10}	0.89	1.08	1.04
8	8.39×10^{-15}	4.19×10^{-12}	1.39×10^{-11}	0.95	1.00	1.00

Conclusion

The aim of this work was to describe the bivariate spectral relaxation method for systems of coupled partial differential equations. This technique extends the previous spectral relaxation method of Motsa et al. (2014a) to allow for discretization of both time and space derivatives using spectral collocation methods. The following conclusions can be drawn regarding this method;

- The method gives accurate results in the whole space and time domains $\xi \in [0, 1]$ and $\tau \in [0, \infty)$ with residual errors rapidly approaching zero.
- The application of spectral collocation to both time and space derivatives ensures that the method performs significantly better than the SRM and the Keller-Box method in terms of computational time.
- The algorithm involves the usage of known formulas for discretization using Chebyshev spectral collocation.
- In future we intend to show that the technique can be extended to coupled nonlinear systems in three-dimensions.

Authors' contributions

All authors participated in the preparation of the manuscript. All authors read and approved the final manuscript.

Author details

[1] School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Private Bag X01, Scottsville, Pietermaritzburg 3209, South Africa. [2] Department of Applied Physics and Engineering Mathematics, University of Johannesburg, P.O. Box 17011, Doornfontein, Johannesburg 2028, South Africa.

Acknowledgements

This work was supported in part by the University of KwaZulu-Natal.

Competing interests

The authors declare that they have no competing interests.

Received: 19 August 2015 Accepted: 24 March 2016
Published online: 14 April 2016

References

Abbas Z, Hayat T, Sajid M, Asghar S (2008) Unsteady flow of a second grade fluid film over an unsteady stretching sheet. Math Comput Model 48:518–526

Ahmad I, Sajid M, Hayat T, Ayub M (2008) Unsteady axisymmetric flow of a second-grade fluid over a radially stretching sheet. Comput Math Appl 56:1351–1357

Ali FM, Nazar R, Arifin NM (2010) Numerical solutions of unsteady boundary layer flow due to an impulsively stretching surface. J Appl Comp Sci Math 8(4):25–30

Ali A, Amin N, Pop I (2010) Unsteady Mixed convection boundary layer from a circular cylinder in a micropolar fluid. Int J Chem Eng. doi:10.1155/2010/417875

Ali A, Mehmood A (2008) Homotopy analysis of unsteady boundary layer flow adjacent to permeable stretching surface in a porous medium. Commun Nonlinear Sci Numer Simul 13:340–349

Alizadeh-Pahlavan A, Sadeghy K (2009) On the use of homotopy analysis method for solving unsteady MHD flow of Maxwellian fluids above impulsively stretching sheets. Commun Nonlinear Sci Numer Simul 14:1355–1365

Cebeci T, Bradshaw P (1984) Physical and computational aspects of convective heat transfer. Springer, New York

Dlamini PG, Motsa SS, Khumalo M (2013) On the comparison between compact finite difference and pseudospectral approaches for solving similarity boundary layer problems. Article ID 746489

Fan T, Xu H, Pop I (2010) Unsteady stagnation flow and heat transfer towards a shrinking sheet. Int Commun Heat Mass Transf 37:1440–1446

Hayat T, Qasim M, Abbas Z (2010) Homotopy solution for the unsteady three-dimensional MHD flow and mass transfer in a porous space. Commun Nonlinear Sci Numer Simul 15:2375–2387

Liao SJ (2012) Homotopy analysis method in nonlinear differential equations. Higher Education Press, Berlin

Lok YY, Amin N, Pop I (2003) Unsteady boundary layer flow of a micropolar fluid near the rear stagnation point of a plane surface. Int J Therm Sci 42(11):995–1001

Mehmood A, Ali A, Shah T (2008) Heat transfer analysis of unsteady boundary layer flow by homotopy analysis method. Commun Nonlinear Sci Numer Simul 13:902–912

Motsa SS, Dlamini PG, Khumalo M (2014a) Spectral relaxation method and spectral quasilinearization method for solving unsteady boundary layer flow problems. Adv Math Phys. doi:10.1155/2014/341964
Motsa SS, Awad FG, Makukula ZG, Sibanda P (2014b) The spectral homotopy analysis method extended to systems of partial differential equations. Abstr Appl Anal 2014:241594-1–241594-11. doi:10.1155/2014/241594
Nazar N, Amin N, Pop I (2004a) Unsteady boundary layer flow due to stretching surface in a rotating fluid. Mech Res Commun 31:121–128
Nazar R, Amin N, Filip D, Pop I (2004b) Unsteady boundary layer flow in the region of the stagnation point on a stretching sheet. Int J Eng Sci 42:1241–1253
Trefethen LN (2000) Spectral methods in MATLAB. SIAM, Philadelphia
Xu H, Liao SJ, Pop I (2007) Series solutions of unsteady three-dimensional MHD flow and heat transfer in the boundary layer over an impulsively stretching plate. Eur J Mech B/Fluids 26:15–27
You X, Xu H, Pop I (2010) Homotopy analysis of unsteady heat transfer started impulsively from rest along a symmetric wedge. Int Commun Heat Mass Transf 37:47–51

Appendix C

MD-BSQLM

This article is about the multi-domain bivariate quasilinearisation method for nonlinear evolution equations. It appears in Chapter 7 of this thesis.

International Journal of Computational Methods
Vol. 14, No. 2 (2017) 1750041 (27 pages)

DOI: 10.1142/S0219876217500414

A Multi-Domain Bivariate Pseudospectral Method for Evolution Equations

V. M. Magagula, S. S. Motsa and P. Sibanda*

School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal, Private Bag X01
Scottsville 3209, Pietermaritzburg, South Africa
**sibandap@ukzn.ac.za*

Received 7 March 2016
Revised 27 May 2016
Accepted 11 September 2016
Published 7 October 2016

In this paper, we present a new general approach for solving nonlinear evolution partial differential equations. The novelty of the approach is in the combination of spectral collocation and Lagrange interpolation polynomials with Legendre–Gauss–Lobatto grid points to descritize and solve equations in piece-wise defined intervals. The method is used to solve several nonlinear evolution partial differential equations, namely, the modified KdV–Burgers equation, modified KdV equation, Fisher's equation, Burgers–Fisher equation, Burgers–Huxley equation and the Fitzhugh–Nagumo equation. The results are compared with known analytic solutions to confirm accuracy, convergence and to get a general understanding of the performance of the method. In all the numerical experiments, we report a high degree of accuracy of the numerical solutions. Strategies for implementing various boundary conditions are discussed.

Keywords: Multi-domain; bivariate interpolation; spectral quasilinearization method; evolution equations; Legendre–Gauss–Lobatto grid points.

1. Introduction

Evolution nonlinear partial differential equations (NPDEs) are useful for modeling many naturally occurring phenomena. There are different variants of evolution NPDE's, such as, for example, NPDEs that model traveling wave fronts which have major applications in chemistry, biology and medicine [Sherratt (1998)]. NPDEs also arise in various science and engineering fields, such as in shock wave formation, turbulence, heat conduction, traffic flow, gas dynamics, sound waves in viscous medium, [Nawaz *et al.* (2013); Mittal and Jiwari (2009a; 2009b); Mittal and Tripathi (2014); Kheiri and Ebadi (2010); Jiwari *et al.* (2012)]).

It is however difficult to obtain analytic solutions of NPDE's due to their nonlinearity and complexities over large time domains. For this reason, there is always a

*Corresponding author.

need to develop numerical methods that are computationally fast, converge quickly and are accurate. Pseudospectral methods have such qualities. They have been used, and often preferred, because they use a few grid points to achieve a high degree of numerical accuracy, and because they use few grid points, they generally require minimal computational time.

Spectral methods with Lagrange interpolation polynomials and Legendre–Gauss–Lobatto grid points have been used to solve various differential equations. For example, they have been used to solve ordinary differential equations [Adibi and Rismani (2010); Rismani and Monfared (2012); Pandey *et al.* (2012)], optimal control problems [Wu and Wang (2009); Elnagar and Kazemi (1997); Fahroo and Ross (2001); Tian and Zong (2010)] and partial differential equations [Wang and Guo (2008); Ma (2001); Wu *et al.* (2003); Li *et al.* (2000); Shamsi and Dehghan (2006); Ma and Sun (2000); Bhrawy (2013); Shamsi and Dehghan (2012); Bhrawy *et al.* (2015)]. Li *et al.* [2000], Shamsi and Dehghan [2006], and Ma and Sun [2000], approximated space derivatives using pseudospectral methods and time derivatives using finite differences when solving NPDEs. Finite differences generally require a large set of grid points for the underlying method to converge. On the other hand, pseudospectral methods are more accurate, and use few Legendre–Gauss–Lobatto grid points to achieve accurate results and hence are computationally faster compared to finite differences. Applying pseudospectral methods in space, and finite differences in time however leads to a slower and less accurate solution.

Motsa *et al.* [2014] used spectral collocation independently in both space and time in order to improve the accuracy and computational speed in numerical simulations. The method introduced in Motsa *et al.* [2014] uses a quasilinearization technique developed by Bellman and Kalaba [1965], the Chebyshev spectral collocation method and bivariate Lagrange interpolation with Chebyshev–Gauss–Lobatto grid points. The numerical method gave accurate results for smaller time domains. The level of accuracy however deteriorated with an increase in the time domain. Also, numerical experiments have shown that using Legendre–Gauss–Lobatto grid points lead to slightly accurate solutions. In this paper, we propose to extend the work of Motsa *et al.* [2014] by using the multi-domain approach in the time variable. This leads to significant improvements in the accuracy of the method for large values of time t. To ensure improved accuracy, we use Legendre–Gauss–Lobatto grid points in the algorithm instead of Chebyshev–Gauss–Lobatto grid points. Linear forms of the evolution partial differential equations are obtained by using the quasilinearization method. The pseudospectral collocation method is applied independently both in space and time variables of the linearized equations. The multi-domain approach is used only in the time variable t which is divided into small nonoverlapping subintervals on which the pseudospectral collocation method is used to solve the partial differential equations. The continuity condition is used to advance the solution across the sub-intervals. A similar approach has been used previously to solve systems of first-order chaotic initial values problems [Dlamini *et al.* (2013); Shateyi

et al. (2014); Motsa *et al.* (2013); Motsa *et al.* (2012); Motsa (2012); Motsa and Sibanda (2012)].

The accuracy, reliability and robustness of the method is tested by solving a number of nonlinear evolution equations, namely the modified KdV–Burgers, the modified KdV, Fishers, Burgers–Fisher, Burgers–Huxley and the Fitzhugh–Nagumo equations. The numerical results are benchmarked against exact solutions. The order of accuracy of the method and the time taken to compute accurate solutions are displayed in tables.

The organization of the paper is as follows. In Sec. 2, we introduce the method for a general nth order nonlinear evolution PDE. In Sec. 3, we describe the application of the method using various boundary conditions. The numerical experiments and results are presented in Secs. 4 and 5, respectively. Finally, we conclude in Sec. 6.

2. A Multi-Domain Bivariate Approach

In this section, the multi-domain based Legendre–Gauss–Lobbatto bivariate Lagrange spectral quasilinearization method for approximating solutions to nonlinear evolution partial differential equations is introduced. Without loss of generality, we consider an nth order NPDE of the form,

$$\frac{\partial u}{\partial \xi} = \mathcal{H}\left(u, \frac{\partial u}{\partial \nu}, \frac{\partial^2 u}{\partial \nu^2}, \ldots, \frac{\partial^n u}{\partial \nu^n}\right), \tag{1}$$

with the physical region $\{(\xi, \nu) \,|\, \xi \in [0, T], \nu \in [a, b]\}$. The constant n denotes the order of differentiation, the required solution is denoted by $u(\nu, \xi)$ and $\mathcal{H}$ is the nonlinear operator which contains $u(\nu, \xi)$ and all the spatial derivatives of $u(\nu, \xi)$. The multi-domain technique approach assumes that the time interval can be decomposed into p nonoverlapping sub-intervals. Let $\xi \in \Gamma$ where $\Gamma = [0, T]$ be the time interval where the solution of general nonlinear parabolic PDE exists. The sub-intervals are defined as

$$\Gamma_l = (\xi_{l-1}, \xi_l), \quad l = 1, 2, \ldots, p \quad \text{with } 0 = \xi_0 < \xi_1 < \xi_2 < \cdots < \xi_p = T. \tag{2}$$

The main task in the multi-domain approach is determining the solution of Eq. (1) independently on each sub-interval, one at a time, beginning at the initial condition. The initial condition is considered to be the left boundary of the time interval. The computed solution in the first interval is used to compute the solutions in the remaining $l-1$ sub-intervals. The computed solution at the right hand boundary of the first interval is used as an initial condition in the subsequent sub-interval. The process is repeated until the last sub-interval. The process of matching the solutions in different intervals along their common boundary is called patching. The patching condition requires that

$$u^{(l)}(\nu, \xi_{l-1}) = u^{(l-1)}(\nu, \xi_{l-1}), \quad \nu \in [a, b], \tag{3}$$

where $u^{(l)}(\nu, \xi)$ denotes the solution of Eq. (1) at each sub-interval Γ_l. Since the grid points and differentiation matrices are defined in the interval $[-1, 1]$, then, in each sub-interval Γ_l, the time interval $\{\xi^l \,|\, \xi_{l-1} \leq \xi^l \leq \xi_l\}$ is transformed to $\{t| -1 \leq t \leq 1\}$ using the linear transformation

$$\xi^l = \frac{1}{2}(\xi_l - \xi_{l-1})t + \frac{1}{2}(\xi_l + \xi_{l-1}). \tag{4}$$

Similarly, the space region, $\{\nu \,|\, a \leq \nu \leq b\}$ is transformed using the linear transformation

$$\nu = \frac{1}{2}(b - a)x + \frac{1}{2}(b + a). \tag{5}$$

to the region $\{x \,|\, -1 \leq x \leq 1\}$. Therefore, in each sub-interval, we are required to solve the nonlinear parabolic equation

$$\frac{\partial u^{(l)}}{\partial t} = H\left(u^{(l)}, \frac{\partial u^{(l)}}{\partial x}, \frac{\partial^2 u^{(l)}}{\partial x^2}, \ldots, \frac{\partial^n u^{(l)}}{\partial x^n}\right), \quad t \in [-1, 1], \quad x \in [-1, 1]. \tag{6}$$

subject to

$$u^{(l)}(x, t_{l-1}) = u^{(l-1)}(x, t_{l-1}), \tag{7}$$

where $l = 1, 2, \ldots, p$. The solution method assumes that the solution in each sub-interval Γ_l, denoted by $u^{(l)}(x, t)$, can be approximated by a bivariate Lagrange interpolation polynomial of the form

$$u^{(l)}(x, t) \approx \sum_{i=0}^{M_x} \sum_{j=0}^{M_t} u^{(l)}(x_i, t_j)\mathcal{L}_i(x)\mathcal{L}_j(t). \tag{8}$$

The bivariate Lagrange interpolation polynomial interpolates $u^{(l)}(x, t)$ at carefully chosen grid points in both x and t directions. The selected grid points in both x and t directions are respectively given by the roots of the polynomials [Canuto *et al.* (1988); Canuto *et al.* (2007); Voigt *et al.* (1984)]

$$p(x) = (1 - x^2)P'_{M_x}(x), \quad p(t) = (1 - t^2)P'_{M_t}(t), \tag{9}$$

where P'_{M_t} is the first derivative of the M_t order Legendre polynomial, which is given by

$$P'_{M_t}(t) = \frac{M_t(M_t + 1)}{(2M_t + 1)}\left[\frac{P_{M_t-1}(t) - P_{M_t+1}(t)}{1 - t^2}\right]. \tag{10}$$

The zeros of $P'_{M_t}(t)$ can be computed using any polynomial root finding method. In this work, the second order Newton's method was used. These grid points are termed the Legendre–Gauss–Lobatto grid points. The function $\mathcal{L}_i(x)$ is the characteristic Lagrange cardinal polynomial based on the Legendre–Gauss–Lobatto points

[Canuto *et al.* (1988); Canuto *et al.* (2007); Voigt *et al.* (1984)]

$$\mathcal{L}_i(x) = \prod_{\substack{k=0 \\ k\neq i}}^{M_x} \frac{x - x_k}{x_i - x_k}. \tag{11}$$

The characteristic Lagrange cardinal polynomial $\mathcal{L}_i(x)$ based on the Legendre–Gauss–Lobatto points, can also be expressed in the form [Canuto *et al.* (1988); Canuto *et al.* (2007); Voigt *et al.* (1984)]

$$\mathcal{L}_i(x) = \frac{1}{M_x(M_x+1)P_{M_x}(x_i)} \frac{(x^2-1)P'_{M_x}(x)}{(x - x_i)}. \tag{12}$$

Equation (6) can be expressed in the form:

$$H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \ldots, u^{(l)}_{(x,n)}\right] - u^{(l)}_{(t,1)} = 0, \tag{13}$$

where $u^{(l)}_{(x,n)}$ denotes the nth partial derivative of $u(x,t)$ with respect to x in the lth sub-interval. Similarly, $u^{(l)}_{(t,1)}$ denotes the first partial derivative of $u(x,t)$ with respect to t in the lth sub-interval and H is the nonlinear operator. We assume that the difference $u^{(l)}_{(x,0,s+1)} - u^{(l)}_{(x,0,s)}$ (note that s and $s+1$ denote previous and current iterations, respectively.) and all it's space derivatives are small. The nonlinear operator H is approximated by using the linear terms of the Taylor series and thus

$$\begin{aligned} &H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \ldots, u^{(l)}_{(x,n)}\right] \\ &\quad\approx H\left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right] \\ &\qquad + \sum_{k=0}^{n} \frac{\partial H}{\partial u^{(l)}_{(x,k)}} \left(u^{(l)}_{(x,k,s+1)} - u^{(l)}_{(x,k,s)}\right). \end{aligned} \tag{14}$$

Let

$$\begin{aligned} &\frac{\partial H}{\partial u^{(l)}_{(x,k)}} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right] \\ &\qquad = \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right]. \end{aligned} \tag{15}$$

Therefore, Eq. (14) can be expressed as

$$\begin{aligned} &H\left[u^{(l)}_{(x,0)}, u^{(l)}_{(x,1)}, u^{(l)}_{(x,2)}, \ldots, u^{(l)}_{(x,n)}\right] \\ &\quad\approx H\left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right] \\ &\qquad + \sum_{k=0}^{n} \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right] u^{(l)}_{(x,k,s+1)} \\ &\qquad - \sum_{k=0}^{n} \omega^{(l)}_{k,s} \left[u^{(l)}_{(x,0,s)}, u^{(l)}_{(x,1,s)}, u^{(l)}_{(x,2,s)}, \ldots, u^{(l)}_{(x,n,s)}\right] u^{(l)}_{(x,k,s)} \end{aligned} \tag{16}$$

Let

$$R_s^{(l)}\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right] = \sum_{k=0}^{n} \omega_{k,s}^{(l)}\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right] u_{(x,k,s)}^{(l)} - H\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right]. \tag{17}$$

Equation (16) can be expressed as

$$H\left[u_{(x,0)}^{(l)}, u_{(x,1)}^{(l)}, u_{(x,2)}^{(l)}, \ldots, u_{(x,n)}^{(l)}\right] \approx \sum_{k=0}^{n} \omega_{k,s}^{(l)}\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right] u_{(x,k,s+1)}^{(l)} - R_s^{(l)}\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right]. \tag{18}$$

Substituting Eq. (18) into Eq. (13), we get

$$\sum_{k=0}^{n} \omega_{k,s}^{(l)} u_{(x,k,s+1)}^{(l)} - u_{(t,1,s+1)}^{(l)} = R_s^{(l)}\left[u_{(x,0,s)}^{(l)}, u_{(x,1,s)}^{(l)}, u_{(x,2,s)}^{(l)}, \ldots, u_{(x,n,s)}^{(l)}\right]. \tag{19}$$

Equation (19) is a linearized form of Eq. (6). The next nontrivial important procedure, termed collocation, is the evaluation of the time derivative at the Legendre–Gauss–Lobatto grid points t_j $(j = 0, 1, \ldots, M_t)$ and the space derivatives at the Legendre–Gauss–Lobatto grid points x_i $(i = 0, 1, \ldots, M_x)$. The values of the time derivatives are computed at the Legendre–Gauss–Lobatto points (x_i, t_j), as (for $j = 0, 1, 2, \ldots, M_t$)

$$\left.\frac{\partial u^{(l)}}{\partial t}\right|_{(x_i,t_j)} = \sum_{\rho=0}^{M_x}\sum_{\eta=0}^{M_t} u^{(l)}(x_\rho, t_\eta)\mathcal{L}_\rho(x_i)\frac{d\mathcal{L}_\eta(t_j)}{dt}, \tag{20}$$

$$= \sum_{\eta=0}^{M_t} u^{(l)}(x_i, t_\eta) d_{j\eta} = \sum_{\eta=0}^{M_t} d_{j\eta} u^{(l)}(x_i, t_\eta), \tag{21}$$

where for $l = 1, 2, \ldots, p$, $d_{j\eta} = \frac{d\mathcal{L}_\eta(t_j)}{dt}$, are the $j\eta$th elements of the standard first derivative Legendre differentiation matrix of size $(M_t + 1) \times (M_t + 1)$. The first derivative Legendre differentiation matrix with respect to the Legendre–Gauss–Lobatto points is given by [Canuto *et al.* (1988); Canuto *et al.* (2007);

Voigt *et al.* (1984)],

$$d_{j\eta} = \begin{cases} -\dfrac{M_t(M_t+1)}{4} & \text{if } \eta = j = 0 \\ \dfrac{P_{M_t}(t_j)}{P_{M_t}(t_\eta)[t_j - t_\eta]} & \text{if } \eta \neq j, 0 \leq j, \eta \leq M_t \\ 0 & \text{if } 1 \leq j = \eta \leq M_t - 1 \\ \dfrac{M_t(M_t+1)}{4} & \text{if } \eta = j = M_t. \end{cases} \tag{22}$$

The values of the space derivatives are computed at the Legendre–Gauss–Lobatto points (x_i, t_j) (for $i = 0, 1, 2, \ldots, M_x$) as

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_i,t_j)} = \sum_{\rho=0}^{M_x}\sum_{\eta=0}^{M_t} u^{(l)}(x_\rho, t_\eta)\frac{d\mathcal{L}_\rho(x_i)}{dx}\mathcal{L}_\eta(t_j), \tag{23}$$

$$= \sum_{\rho=0}^{M_x} u^{(l)}(x_\rho, t_j) D_{i\rho} = \sum_{\rho=0}^{M_x} D_{i\rho} u^{(l)}(x_\rho, t_j), \tag{24}$$

where $D_{i\rho} = \frac{d\mathcal{L}_\rho(x_i)}{dx}$, are the $i\rho$th elements of the standard first derivative Legendre differentiation of size $(M_x + 1) \times (M_x + 1)$ as defined by Eq. (22). Similarly, the nth order derivative is defined as

$$\left.\frac{\partial^n u^{(l)}}{\partial x^n}\right|_{(x_i,t_j)} = \sum_{\rho=0}^{M_x} D_{i\rho}^n u^{(l)}(x_\rho, t_j) = \left[\mathbf{D}^n \mathbf{U}_j^{(l)}\right]_i, \quad i = 0, 1, 2, \ldots, M_x, \tag{25}$$

where $[\mathbf{D}^n \mathbf{U}_j^{(l)}]_i$ denotes the ith element of the column matrix $\mathbf{D}^n \mathbf{U}_j^{(l)}$. The vector $\mathbf{U}_j^{(l)}$ is defined as

$$\mathbf{U}_j^{(l)} = [u^{(l)}(x_0, t_j), u^{(l)}(x_1, t_j), \ldots, u^{(l)}(x_{M_x}, t_j)]^T. \tag{26}$$

The superscript T in Eq. (26) denotes matrix transpose. If we evaluate Eq. (19) at each grid point (x_i, t_j) and substitute Eq. (21) and (25) we get

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^k \mathbf{U}_{s+1,j}^{(l)} - \sum_{k=0}^{M_t} d_{jk} \mathbf{U}_{s+1,k}^{(l)} = R_s^{(l)}, \tag{27}$$

for $j = 0, 1, 2, \ldots, M_t$, where $\mathbf{\Omega}_{k,r}$ is a diagonal matrix given by:

$$\mathbf{\Omega}_{k,r} = \begin{bmatrix} \omega_{k,s}(x_0, t_j) & & & \\ & \omega_{k,s}(x_1, t_j) & & \\ & & \ddots & \\ & & & \omega_{k,s}(x_{M_x}, t_j) \end{bmatrix}. \tag{28}$$

Since the initial condition for Eq. (27) corresponds to $\xi_{M_t} = -1$, we express Eq. (27) as

$$\sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^k \mathbf{U}_{s+1,j}^{(l)} - \sum_{k=0}^{M_t-1} d_{jk} \mathbf{U}_{s+1,k}^{(l)} = \mathbf{R}_j^{(l)}, \tag{29}$$

where

$$\mathbf{R}_j^{(l)} = R_s^{(l)} + d_{jM_t} \mathbf{U}_{M_t}^{(l)}, \quad \text{for } j = 0, 1, 2, \ldots, M_t - 1.$$

For $j = 0, 1, 2, \ldots, M_t - 1$, Eq. (29) forms an $M_t(M_x + 1) \times M_t(M_x + 1)$ matrix equation

$$\begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,M_t-1} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,M_t-1} \\ \vdots & \vdots & \ddots & \vdots \\ A_{M_t-1,0} & A_{M_t-1,1} & \cdots & A_{M_t-1,M_t-1} \end{bmatrix} \begin{bmatrix} \mathbf{U}_0^{(l)} \\ \mathbf{U}_1^{(l)} \\ \vdots \\ \mathbf{U}_{M_t-1}^{(l)} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_0^{(l)} \\ \mathbf{R}_1^{(l)} \\ \vdots \\ \mathbf{R}_{M_t-1}^{(l)} \end{bmatrix}, \tag{30}$$

where

$$A_{i,i} = \sum_{k=0}^{n} \mathbf{\Omega}_{k,s} \mathbf{D}^{(k)} - d_{i,i} \mathbf{I}, \tag{31}$$

$$A_{i,j} = -d_{i,j} \mathbf{I}, \quad \text{when } i \neq j, \tag{32}$$

and $\mathbf{I}$ is the identity matrix of size $(M_x + 1) \times (M_x + 1)$. Solving Eq. (29) gives $u^{(l)}(x_i, t_j)$ which is subsequently used in Eq. (8) to approximate $u^{(l)}(x, t)$.

3. Boundary Conditions

In this section, different types of boundary conditions are considered. We consider nonhomogeneous Dirichlet boundary conditions, nonhomogeneous Neumann boundary conditions and nonhomogeneous mixed boundary conditions.

3.1. *Nonhomogeneous dirichlet boundary conditions*

In this section, we consider boundary conditions of the form

$$u(a, t) = f_1(t), \tag{33}$$

$$u(b, t) = f_2(t), \tag{34}$$

in the interval $[a, b]$, where $f_1(t)$ and $f_2(t)$ are nonzero functions. In implementing nonhomogeneous Dirichlet boundary conditions, we note that for $\nu \in [a, b]$, $\nu = a$

corresponds to $x = -1$, for $x \in [-1, 1]$ and hence $u(a, t) = u(\nu(-1), t(\xi))$. Similarly, for $\nu \in [a, b]$, $\nu = b$ corresponds to $x = 1$, for $x \in [-1, 1]$ and hence $u(b, t) = u(\nu(1), t(\xi))$. Therefore, when evaluating the boundary conditions at the Legendre–Gauss–Lobatto grid points in each sub-interval

$$\Gamma_l = (t_{l-1}, t_l), \quad l = 1, 2, \ldots, p, \quad \text{with, } 0 = t_0 < t_1 < t_2 < \cdots < t_p = T, \tag{35}$$

we obtain

$$u^{(l)}(a, t_j) = u^{(l)}(x_{M_x}, t_j) = f_1^{(l)}(t_j), \tag{36}$$

$$u^{(l)}(b, t_j) = u^{(l)}(x_0, t_j) = f_2^{(l)}(t_j), \tag{37}$$

for $j = 0, 1, \ldots, M_t$.

3.2. *Nonhomogeneous neumann boundary conditions*

In this section, we consider boundary conditions of the form

$$\left.\frac{\partial u}{\partial x}\right|_{(a,t)} = g_1(t), \tag{38}$$

$$\left.\frac{\partial u}{\partial x}\right|_{(b,t)} = g_2(t), \tag{39}$$

in the interval $[a, b]$, where $g_1(t)$ and $g_2(t)$ are nonzero functions. In general, the first derivative with respect to x at any grid point (x_i, t_j) is given by Eq. (24). Thus, the nonhomogeneous Neumann boundary conditions can be expressed as

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(a,t_j)} = \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_{M_x},t_j)} = \sum_{\rho=0}^{M_x} D_{M_x\rho} u^{(l)}(x_\rho, t_j) = g_1(t_j), \tag{40}$$

$$\left.\frac{\partial u^{(l)}}{\partial x}\right|_{(b,t_j)} = \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(x_0,t_j)} = \sum_{\rho=0}^{M_x} D_{0\rho} u^{(l)}(x_\rho, t_j) = g_2(t_j), \tag{41}$$

for $j = 0, 1, \ldots, M_t$, respectively.

3.3. *Nonhomogeneous mixed boundary conditions*

In this section, we consider boundary conditions of the form

$$u(a, t) + \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(a,t_j)} = h_1(t), \tag{42}$$

$$u(b, t) + \left.\frac{\partial u^{(l)}}{\partial x}\right|_{(b,t_j)} = h_2(t), \tag{43}$$

in the interval $[a, b]$, where $h_1(t)$ and $h_2(t)$ are nonzero functions. Using the ideas from the previous two sections, Eq. (42) and (43) can be expressed as

$$u^{(l)}(x_{M_x}, t_j) + \sum_{\rho=0}^{M_x} D_{M_x\rho} u^{(l)}(x_\rho, t_j) = h_1(t_j), \tag{44}$$

$$u^{(l)}(x_0, t_j) + \sum_{\rho=0}^{M_x} D_{0\rho} u^{(l)}(x_\rho, t_j) = h_2(t_j), \tag{45}$$

for $j = 0, 1, \ldots, M_t$, respectively.

4. Numerical Experiments

In this section, we apply the proposed algorithm to the popular nonlinear partial differential equations of the form (6) with exact solutions. These examples were considered as numerical experiments [Motsa *et al.* (2014)]. In all our calculations, we consider nonhomogeneous Dirichlet boundary conditions. The space and time domains are given by $[a, b] = [0, 5]$ and $[t_0, T] = [0, 10]$ respectively for most of the numerical experiments. We choose a $T = 10$ to show the accuracy of the algorithm over a large time domain.

Example 1. We first consider Fisher's equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \alpha u(1 - u), \tag{46}$$

subject to the initial condition

$$u(x, 0) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}\, x}\right)^2}, \tag{47}$$

and exact solution [Wazwaz and Gorguis (2004)]

$$u(x, t) = \frac{1}{\left(1 + e^{\sqrt{\alpha/6}\, x - 5\alpha t/6}\right)^2}, \tag{48}$$

where α is a constant. The Fisher equation represents a reactive-diffusive system and is encountered in chemical kinetics and population dynamics applications. The Fisher's equation has been solved by [Mickens (1997); Olmos *et al.* (2006); Hariharan *et al.* (2009)] using nonstandard finite differences, spectral collocation and Haar wavelet method, respectively. For this example, the appropriate nonlinear operator H is chosen as

$$H(u) = u'' + \alpha u - \alpha u^2, \tag{49}$$

where here, and in the subsequent examples, the primes denote differentiation with respect to x. We use $[a, b] = [0, 5]$ and $[t_0, T] = [0, 10]$ as our x and t domains, respectively.

Example 2. We consider the generalized Burgers–Fisher equation [Golbabai and Javidi (2005)]

$$\frac{\partial u}{\partial t} + \alpha u^{\delta}\frac{\partial u}{\partial x} = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta}), \tag{50}$$

with initial condition

$$u(x,0) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}x\right)\right\}^{\frac{1}{\delta}}, \tag{51}$$

and exact solution

$$u(x,t) = \left\{\frac{1}{2} + \frac{1}{2}\tanh\left(\frac{-\alpha\delta}{2(\delta+1)}\left[x - \left(\frac{\alpha}{\delta+1} + \frac{\beta(\delta+1)}{\alpha}\right)t\right]\right)\right\}^{\frac{1}{\delta}}, \tag{52}$$

where α, β and δ are parameters. For illustration purposes, these parameters are chosen to be $\alpha = \beta = \delta = 1$ in this paper. The Burgers–Fisher equation has been solved by [Javidi *et al.* (2006); Moghimi and Hejazi (2007); Golbabai and Javidi (2009)] using spectral collocation method, spectral domain decomposition and Homotopy analysis method. The nonlinear operator H is chosen as

$$H(u) = u'' + u - uu' - u^2. \tag{53}$$

We use $[a,b] = [0,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively.

Example 3. In this example, the Fitzhugh–Nagumo equation is considered,

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + u(u-\alpha)(1-u) \tag{54}$$

with initial condition

$$u(x,0) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}}\right)\right]. \tag{55}$$

This equation has the exact solution [Li and Guo (2006)]

$$u(x,t) = \frac{1}{2}\left[1 - \coth\left(-\frac{x}{2\sqrt{2}} + \frac{2\alpha-1}{4}t\right)\right], \tag{56}$$

where α is a parameter. In this example, the nonlinear operator H is chosen as

$$H(u) = u'' - \alpha u + (1+\alpha)u^2 - u^3. \tag{57}$$

We use $[a,b] = [1,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively. This equation has been solved by various researchers [Dehghan *et al.* (2010); Abbasbandy (2008); Van Gorder and Vajravelu (2010); Van Gorder (2012)] using the homotopy perturbation method (HPM), the variational iteration method (VIM), the adomian decomposition method (ADM), and homotopy analysis method (HAM).

Example 4. Consider the Burgers–Huxley equation

$$\frac{\partial u}{\partial t} + \alpha u^{\delta} u_x = \frac{\partial^2 u}{\partial x^2} + \beta u(1-u^{\delta})(u^{\delta}-\gamma), \tag{58}$$

where $\alpha, \beta \geq 0$ are constant parameters, δ is a positive integer (set to be $\delta = 1$ in this study) and $\gamma \in (0,1)$. The exact solution subject to the initial condition

$$u(x,0) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r-\alpha}x\right], \tag{59}$$

is reported in Fan [2002] as

$$u(x,t) = \frac{1}{2} - \frac{1}{2}\tanh\left[\frac{\beta}{r-\alpha}(x-ct)\right], \tag{60}$$

where

$$r = \sqrt{\alpha^2 + 8\beta} \quad \text{and} \quad c = \frac{(\alpha - r)(2\gamma - 1) + 2\alpha}{4}. \tag{61}$$

The general solution (60) was also reported in [Hashim *et al.* (2006); Wang *et al.* (1990)]. The Burgers–Huxley has been solved by [Batiha *et al.* (2008); Darvishi *et al.* (2008); Dehghan *et al.* (2012)] using the VIM, spectral collocation and finite differences, respectively. In this example, the nonlinear operator H is chosen as

$$H(u) = u'' - \beta\gamma u - \alpha u u' + \beta(1+\gamma)u^2 - \beta u^3. \tag{62}$$

We use $[a,b] = [0,5]$ and $[t_0, T] = [0,10]$ as our x and t domains, respectively.

Example 5. We consider the modified KdV–Burgers equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} - \frac{\partial^2 u}{\partial x^2} - 6u^2\frac{\partial u}{\partial x}, \tag{63}$$

subject to the initial condition

$$u(x,0) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6}\right), \tag{64}$$

and exact solution [Helal and Mehanna (2006)]

$$u(x,t) = \frac{1}{6} + \frac{1}{6}\tanh\left(\frac{x}{6} - \frac{t}{27}\right). \tag{65}$$

The modified KdV–Burgers equation describes various kinds of phenomena such as a mathematical model of turbulence [Burgers (1948)] and the approximate theory of flow through a shock wave traveling in viscous fluid [Cole (1951)]. It has been solved by [Darvishi *et al.* (2007); Kurulay and Bayram (2010)] using the spectral collocation and by the differential transform method. For this example, the nonlinear operator H is chosen as

$$H(u) = u''' - u'' - 6u'u^2. \tag{66}$$

We use $[a,b] = [0,5]$ and $[t_0, T] = [0,10]$ as our x and t domains, respectively.

Example 6. We consider the nonlinear modified KdV equation

$$\frac{\partial u}{\partial t} = \frac{\partial^3 u}{\partial x^3} + \left(\frac{\partial u}{\partial x}\right)^2 - u^2, \tag{67}$$

subject to the initial condition

$$u(x,0) = \frac{1}{2} + \frac{e^{-x}}{4} \tag{68}$$

and exact solution [Polyanin and Zaitsev (2004)]

$$u(x,t) = \frac{1}{t+2} + \frac{e^{-(x+t)}}{(t+2)^2}. \tag{69}$$

For this example, the nonlinear operator H is chosen as

$$H(u) = u''' - (u')^2 - u^2. \tag{70}$$

We use $[a,b] = [0,5]$ and $[t_0,T] = [0,10]$ as our x and t domains, respectively. This problem was solved by [Motsa *et al.* (2014)] using the bivariate spectral quasilinearization method.

5. Results and Discussion

In this section, we discuss and present the numerical results. The results were all generated using MATLAB 2013. To compare the accuracy, computational time, and general performance of the method, we compared the maximum error terms and the convergence of the method. In order to determine the level of accuracy of the approximate solution, at a particular time level in comparison with the exact solution, we report maximum errors defined by

$$E_{M_x} = \max_k \{|u_e(x_k,t) - u_a(x_k,t)|, : 0 \le k \le M_x\}, \tag{71}$$

where $u_a(x_k,t)$ is the approximate solution and $u_e(x_k,t)$ is the exact solution at the time level t. Here, we have used $p = 10$ and varying values of M_t and M_x, determined through numerical experimentation. For convenience, we refer to the current method using the abbreviations MD-LGL-BSQLM and the earlier method of Motsa *et al.* [2014] by LGL-BSQLM.

5.1. *Maximum error estimates*

In this section, we analyze the accuracy and computational time of the LGL-BSQLM and MD-LGL-BSQLM methods. We report the maximum errors obtained when solving the nonlinear evolution partial differential equations using both methods. We also report on the central processing unit (CPU) time taken to approximate the solutions of the nonlinear evolution partial differential equations. In all cases, we used $M_t = 10$ and varying values of M_x.

In Table 1, the maximum error estimates for the Fisher equation are displayed. We observe that the maximum errors obtained using the LGL-BSQLM method are

Table 1. Maximum error estimates E_{M_x} for the Fishers equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	1.840e-03	1.842e-03	9.508e-06	7.19e-10
2	2.906e-03	2.960e-03	1.033e-05	6.82e-10
3	1.150e-03	1.163e-03	4.555e-06	2.86e-10
4	7.918e-04	7.918e-04	2.597e-06	5.21e-11
5	5.383e-04	5.389e-04	5.600e-07	3.77e-12
6	3.721e-04	3.721e-04	3.176e-07	2.23e-12
7	2.494e-04	2.494e-04	6.682e-08	1.37e-13
8	1.172e-04	1.206e-04	1.077e-08	5.40e-14
9	1.264e-04	1.268e-04	9.716e-09	1.80e-14
10	2.210e-05	2.483e-05	6.086e-09	4.44e-15
CPU time	0.015507	0.044061	0.005363	0.009849

bigger compared to those obtained using the MD-LGL-BSQLM method. This suggests that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve spectrally accurate results. On average, for the same values of M_x and M_t, the difference between maximum error estimates obtained using the MD-LGL-BSQLM and LGL-BSQLM method is of order ten. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 1.

Table 2, shows the maximum error estimates for the Burgers–Fisher equation. The maximum error estimates obtained using the MD-LGL-BSQLM method are smaller compared to those obtained using the LGL-BSQLM method. Therefore,

Table 2. Maximum error estimates E_{M_x} for the Burgers–Fisher equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	1.779e-03	1.772e-03	1.006e-05	3.68e-09
2	2.054e-03	2.066e-03	1.570e-05	7.78e-10
3	2.977e-03	2.977e-03	8.765e-06	4.55e-11
4	2.366e-03	2.366e-03	1.343e-06	3.14e-11
5	5.866e-04	5.856e-04	4.678e-07	4.31e-12
6	1.392e-03	1.392e-03	2.175e-07	5.53e-13
7	1.135e-03	1.135e-03	7.135e-08	1.37e-14
8	4.054e-04	4.054e-04	2.135e-08	1.20e-14
9	1.920e-04	1.923e-04	6.240e-09	4.66e-15
10	9.550e-05	1.034e-04	1.803e-09	4.22e-15
CPU time	0.025428	0.066148	0.011777	0.024576

we conclude that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses fewer grid points to achieve spectral accurate results. Increasing $M_x = 10$ gives an error of approximately 10^{-12} in the time domain $[0, 10]$. Increasing the number of space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 2. Solutions of the Burgers–Fisher equation were obtained in a fraction of a second.

Table 3, shows the maximum error estimates for the Fitzhugh–Nagumo equation, obtained using both the MD-LGL-BSQLM and LGL-BSQLM methods. The maximum errors obtained using the MD-LGL-BSQLM method are smaller compared to those obtained using the LGL-BSQLM method. Therefore, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We also observe that using $M_x = 5$ for the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. Using $M_x = 5$ reduces the size of the variable matrix and in turn reduces computational time. Thus, the MD-LGL-BSQLM method uses fewer grid points to achieve more accurate results for large time domains. Increasing $M_x = 10$ gives an error of approximately 10^{-14} in the time domain $[0, 10]$. Increasing the number of space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 3. Solutions of the Fitzhugh–Nagumo equation were obtained using minimal computational time.

Table 4 displays the maximum error estimates for the Burgers–Huxley equation which were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. It is evident from Table 4 that the maximum errors obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM

Table 3. Maximum error estimates E_{M_x} for the Fitzhugh–Nagumo equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	1.770e-04	1.770e-04	6.049e-07	5.978e-11
2	2.383e-04	2.383e-04	6.071e-08	4.030e-13
3	4.189e-04	4.189e-04	1.037e-08	1.664e-14
4	3.305e-04	3.305e-04	2.430e-09	6.710e-15
5	1.480e-05	1.481e-05	7.108e-10	2.942e-15
6	2.413e-04	2.413e-04	2.455e-10	3.842e-15
7	2.198e-04	2.198e-04	9.683e-11	2.179e-15
8	8.642e-05	8.642e-05	4.264e-11	1.188e-15
9	4.103e-05	4.103e-05	2.056e-11	3.201e-16
10	1.595e-05	1.593e-05	1.065e-11	4.352e-16
CPU time	0.064019	0.076709	0.013449	0.026672

Table 4. Maximum error estimates E_{M_x} for the Burgers–Huxley equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	8.727e-04	8.705e-04	6.245e-17	1.735e-17
2	1.943e-03	1.950e-03	4.580e-16	9.021e-17
3	1.746e-03	1.773e-03	5.135e-16	4.857e-16
4	1.820e-03	1.820e-03	1.110e-15	9.437e-16
5	8.588e-04	8.507e-04	4.441e-15	4.441e-16
6	1.824e-03	1.824e-03	8.882e-16	3.109e-15
7	1.479e-03	1.479e-03	6.439e-15	1.332e-15
8	4.790e-04	4.790e-04	2.220e-15	4.885e-15
9	2.396e-04	2.556e-04	1.288e-14	3.997e-15
10	1.586e-04	1.599e-04	3.220e-15	6.661e-16
CPU time	0.040081	0.084866	0.013389	0.019958

method. Thus, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. For $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. On average, for the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately 12. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 4. The solutions of the Burgers–Huxley equation were also obtained in a fraction of a second.

Table 5 shows the maximum error estimates for the KdV–Burgers equation which were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. In this table, the maximum error estimates obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. Thus,

Table 5. Maximum error estimates E_{M_x} for the KdV-Burgers equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	3.190e-04	3.182e-04	9.086e-07	2.114e-10
2	2.913e-04	2.918e-04	4.190e-07	4.539e-11
3	6.550e-04	6.550e-04	2.285e-07	1.660e-11
4	6.249e-04	6.249e-04	1.097e-07	6.400e-12
5	1.319e-04	1.320e-04	5.032e-08	2.561e-12
6	4.568e-04	4.568e-04	2.284e-08	1.174e-12
7	3.446e-04	3.446e-04	1.038e-08	5.967e-13
8	6.861e-05	7.126e-05	4.743e-09	3.062e-13
9	5.261e-05	5.266e-05	2.185e-09	1.230e-13
10	4.016e-05	4.714e-05	1.016e-09	5.757e-14

the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We observe that for $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. For the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately 11. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 5. The solutions of the Burgers–Huxley equation were also obtained in a fraction of a second.

Table 6 shows the maximum error estimates for the modified KdV equation that were obtained using the MD-LGL-BSQLM and LGL-BSQLM methods. The maximum error estimates obtained using the LGL-BSQLM method are bigger compared to those obtained using the MD-LGL-BSQLM method. Therefore, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. We observe that for $M_x = 5$, the MD-LGL-BSQLM method gives more accurate results than the LGL-BSQLM method with $M_x = 10$ and $M_x = 20$. This in turn reduces computational time and hence the MD-LGL-BSQLM method uses few grid points to achieve more accurate results. For the same values of M_x and M_t, the difference between maximum errors obtained by the MD-LGL-BSQLM and LGL-BSQLM method is approximately 12. Increasing space grid points for the LGL-BSQLM method does not improve the accuracy of the method as shown in Table 6. The solutions of the Burgers–Huxley equation were also obtained in a fraction of a second.

5.2. *Comparison of approximate and exact solutions*

In this section, we consider the approximate and analytical solutions of the nonlinear partial differential equations considered in this work. The graphs show the solutions

Table 6. Maximum error estimates E_{M_x} for the modified KdV equation, with $M_t = 10$.

	LGL-BSQLM		MD-LGL-BSQLM	
$t \backslash M_x$	10	20	5	10
1	4.600e-04	4.794e-04	1.278e-06	4.53e-10
2	2.859e-04	2.859e-04	6.520e-07	1.30e-10
3	5.044e-04	5.044e-04	3.413e-07	4.66e-11
4	3.985e-04	3.985e-04	1.641e-07	1.76e-11
5	8.550e-05	8.785e-05	7.270e-08	7.09e-12
6	2.911e-04	2.911e-04	3.223e-08	2.96e-12
7	2.651e-04	2.651e-04	1.490e-08	1.34e-12
8	1.042e-04	1.042e-04	6.970e-09	5.31e-13
9	4.946e-05	4.946e-05	3.183e-09	2.72e-13
10	5.541e-05	5.556e-05	1.455e-09	5.01e-13
CPU times	0.023869	0.087991	0.006778	0.011656

of the nonlinear partial differential equations in both space and time domains. Figures 1–6 show the approximate and exact solutions of Fisher's equation, Burgers–Fisher equation, Burgers–Huxley equation, Fitzhugh–Nagumo equation, modified KdV–Burgers equation and modified KdV equation, respectively. The approximate solutions in these graphs were generated using the MD-LGL-BSQLM method. The graphs show that the approximate and exact solutions are in excellent agreement in the given time domain for all the equations considered. This implies that the MD-LGL-BSQLM method can be used to approximate solutions of nonlinear partial differential equations in large time domains. These graphs were generated using $M_x = 40$ and $M_t = 10$. The time and space intervals used are $t \in [0, 10]$ and $x \in [0, 5]$ for all the equations.

5.3. *Convergence graphs*

In this section, we compare the convergence and accuracy of the LGL-BSQLM and MD-LGL-BSQLM methods graphically. In generating the results, we used

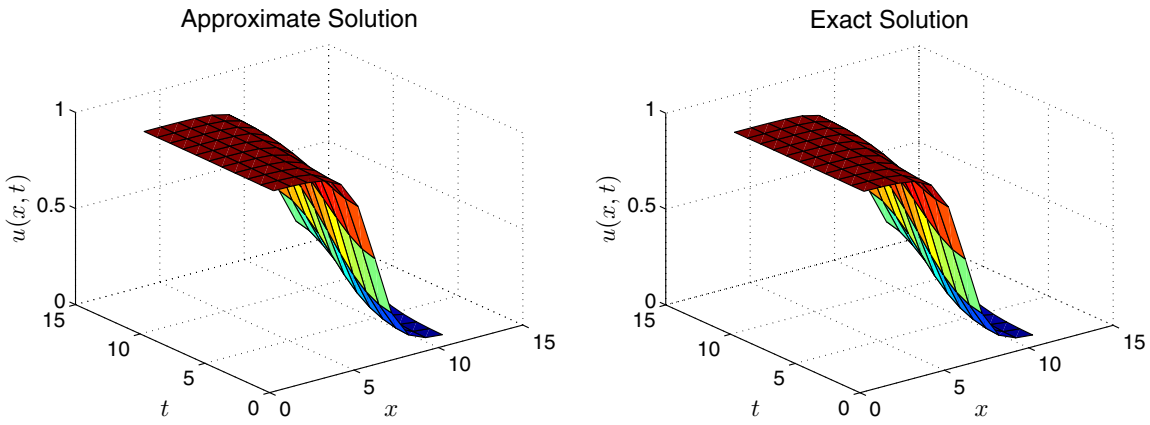

Fig. 1. Approximate and exact solutions of the Fishers equation.

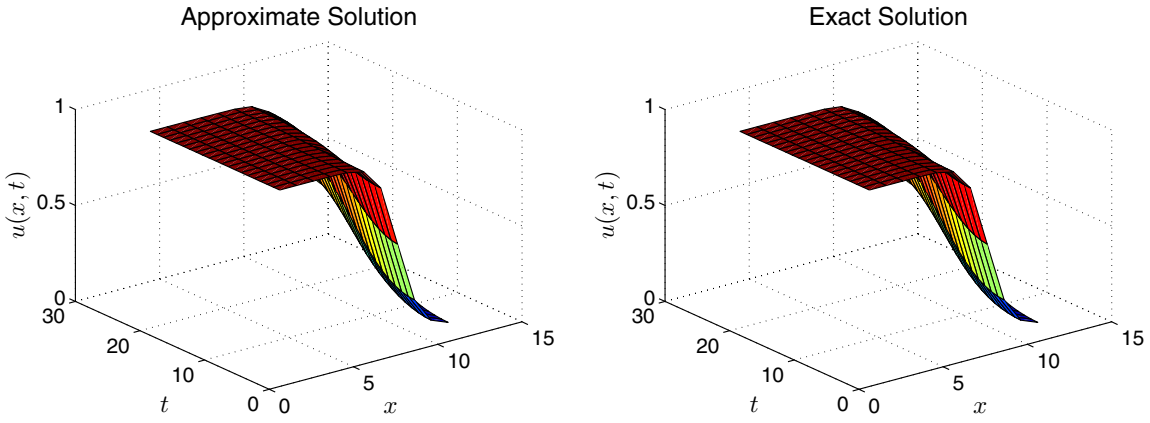

Fig. 2. Approximate and exact solutions of the Burgers–Fisher equation.

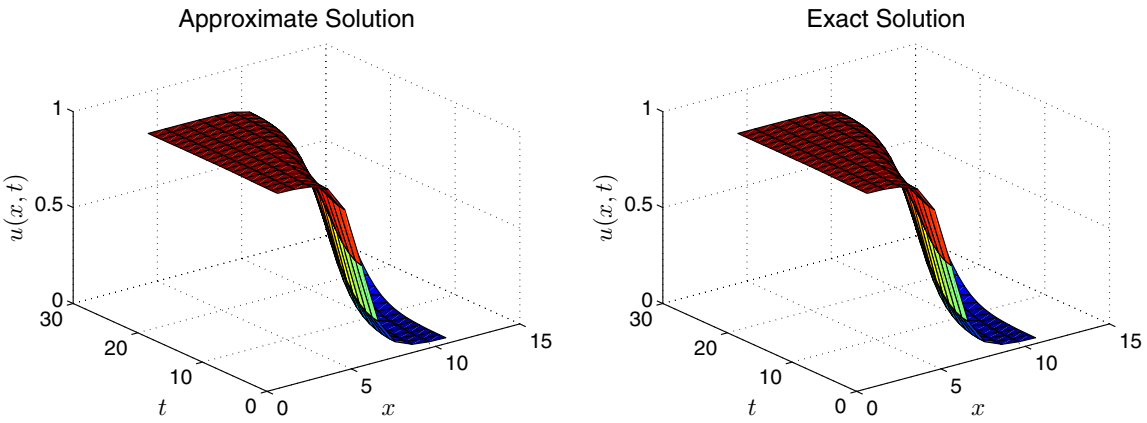

Fig. 3. Approximate and exact solutions of the Burgers–Huxley equation.

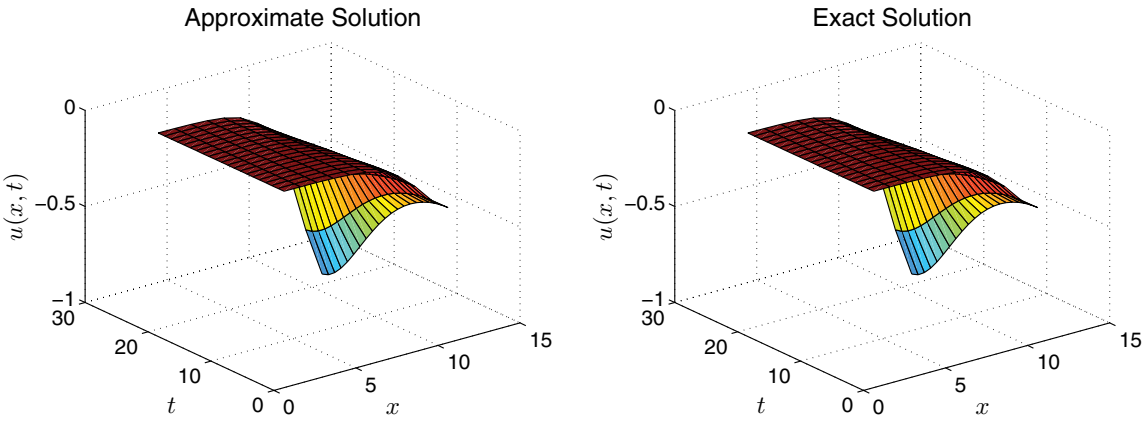

Fig. 4. Approximate and exact solutions of the Fitzhugh–Nagumo equation.

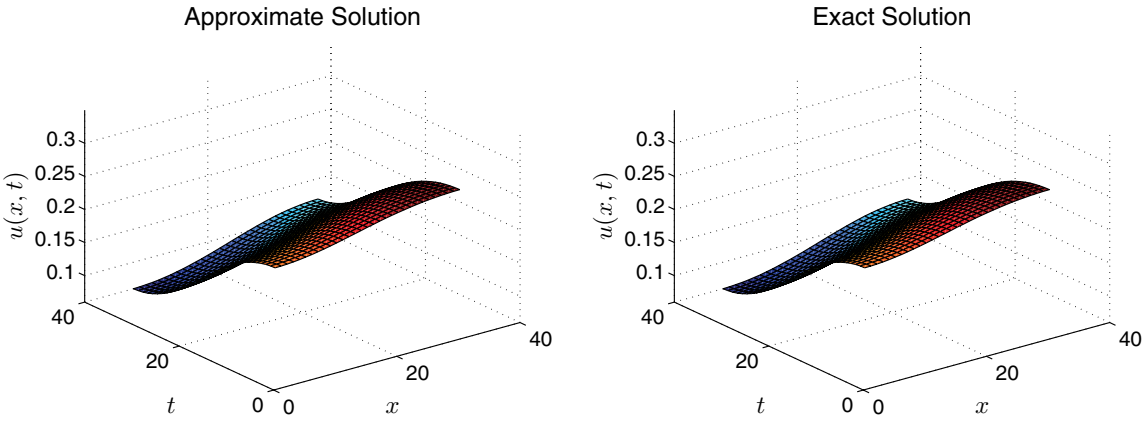

Fig. 5. Approximate and exact solutions of the modified KdV–Burgers equation.

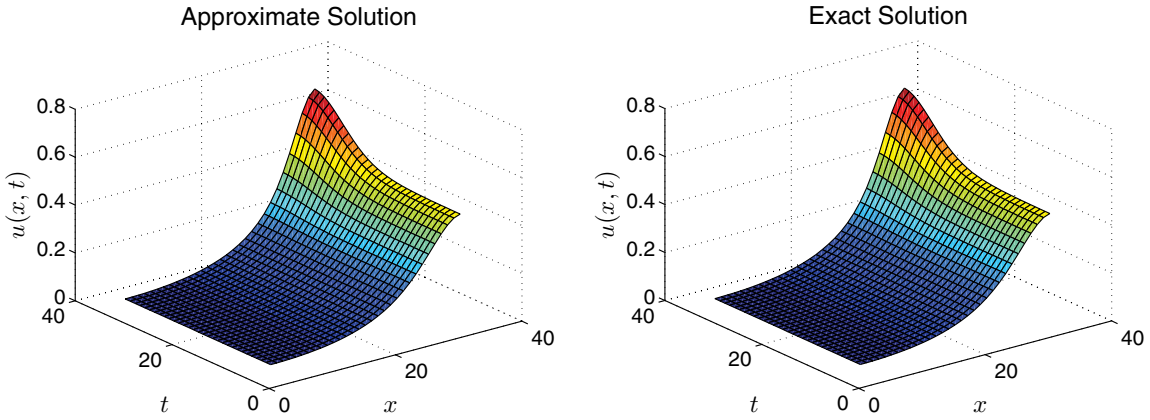

Fig. 6. Approximate and exact solutions of the modified KdV equation.

$M_x = M_t = 10$, $t \in [0, 10]$, and $x \in [0, 5]$. In general, the MD-LGL-BSQLM method converge approximately after two iterations, to a smaller error compared to the LGL-BSQLM method. This implies that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method (see Figs. 7–12).

Figure 7 compares the convergence of the Fisher's equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. The MD-LGL-BSQLM method converges approximately after two iterations to an error of 10^{-14}, while the LGL-BSQLM method converges to 10^{-5} after nine iterations. This implies that the MD-LGL-BSQLM method is a suitable method to approximate the solution of the Fisher's equation for large time domains since it gives more accurate results and converges faster to the exact solution compared to the LGL-BSQLM method.

Figure 8 compares the convergence of the Burgers–Fisher equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. In this case, the MD-LGL-BSQLM method converges to 10^{-12} approximately after two iterations, while the LGL-BSQLM method converges to 10^{-4} after seven iterations. Since the MD-LGL-BSQLM method gives more accurate results and converges faster to the exact

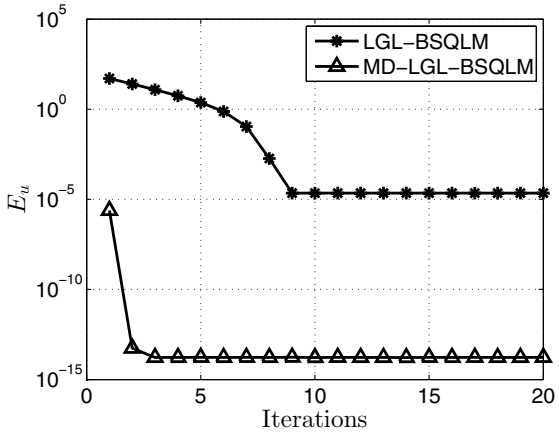

Fig. 7. Fisher's equation convergence graph.

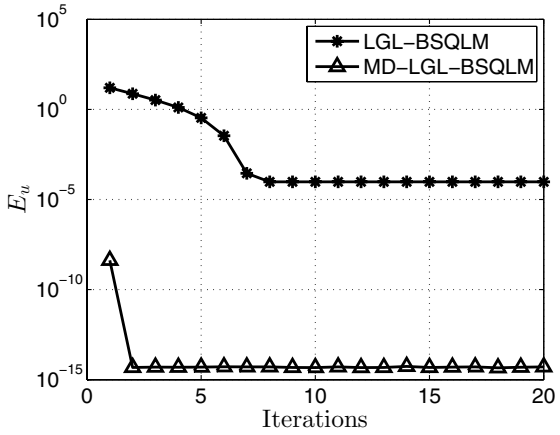

Fig. 8. Burgers–Fisher equation convergence graph.

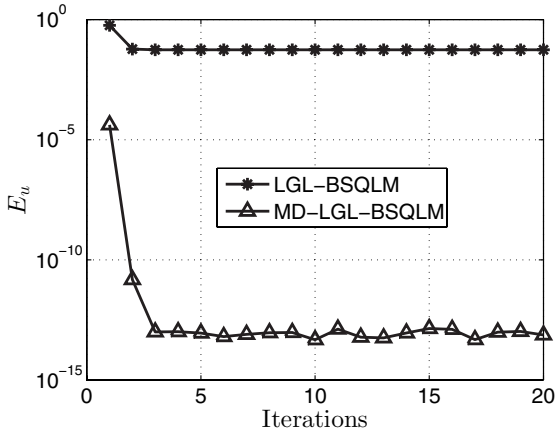

Fig. 9. Fitzhugh–Nagumo equation convergence graph.

solution than the LGL-BSQLM method, then it is a suitable method to approximate the solution of the Burgers–Fisher equation for large time domains.

Figure 9 compares the convergence of the Fitzhugh–Nagumo equation using both the LGL-BSQLM and MD-LGL-BSQLM methods. We observe that the MD-LGL-BSQLM method converges to 10^{-13} after three iterations, while the LGL-BSQLM method converges to 10^{-1} after two iterations. The LGL-BSQLM method converges to a less accurate solution compared to the MD-LGL-BSQLM method. This implies that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method and hence suitable for approximating the solution of the Fitzhugh–Nagumo equation for large time intervals.

In Figure 10, the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of the Burgers–Huxley equation is compared. The graph shows that the MD-LGL-BSQLM method converges to 10^{-12} after two iterations, while the LGL-BSQLM method converges to 10^{-6} after five iterations. Clearly, the LGL-BSQLM method converges to a less accurate solution compared to

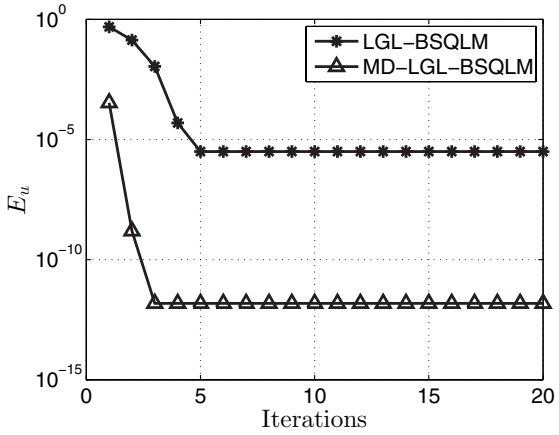

Fig. 10. Burgers–Huxley equation convergence graph.

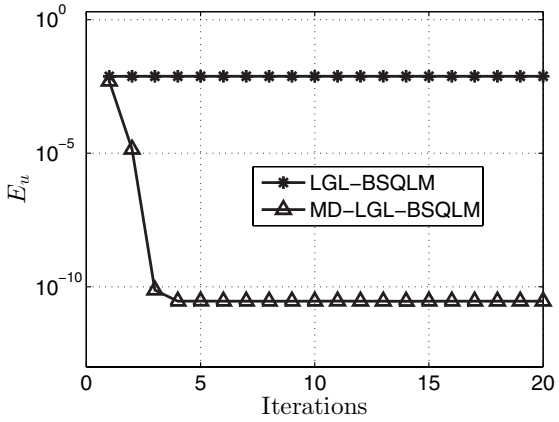

Fig. 11. Modified KdV–Burgers equation convergence graph.

the MD-LGL-BSQLM method. Thus, the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method and hence it is suitable for approximating the solution of the Fitzhugh–Nagumo equation for large time intervals.

Figure 11 compares the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of modified KdV–Burgers equation. The MD-LGL-BSQLM method converges to 10^{-11} after four iterations, while the LGL-BSQLM method converges to 10^{-2} after two iterations. We observe that the LGL-BSQLM method converges to a less accurate solution compared to the MD-LGL-BSQLM method and hence the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. This implies that the MD-LGL-BSQLM method is suitable for approximating solutions of the modified KdV–Burgers equation for large time intervals.

Lastly, Fig. 12 compares the convergence of the LGL-BSQLM and MD-LGL-BSQLM methods for approximating the solution of modified KdV equation. The MD-LGL-BSQLM method converges to 10^{-13} after two iterations, while the

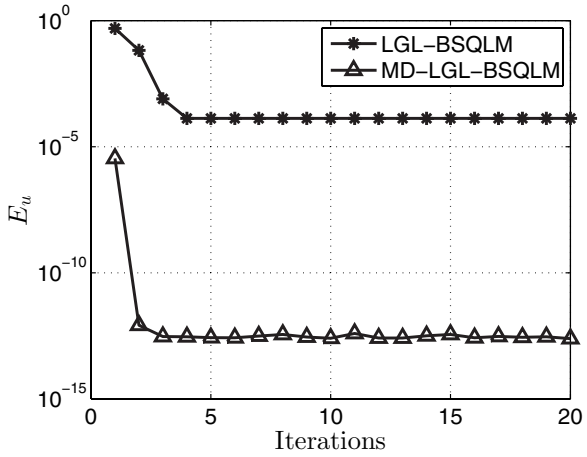

Fig. 12. Modified KdV equation convergence graph.

LGL-BSQLM method converges to 10^{-4} after four iterations. We observe that the MD-LGL-BSQLM method converges to a more accurate solution compared to the LGL-BSQLM method and hence the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method. Thus, the MD-LGL-BSQLM method is suitable for approximating solutions of the modified KdV equation for large time intervals.

In summary, the convergence graphs show that the MD-LGL-BSQLM method converges to a more accurate solution than the LGL-BSQLM method. We can conclude that the MD-LGL-BSQLM method is more accurate than the LGL-BSQLM method for solving nonlinear evolution partial differential equations in large time domains.

6. Conclusion

In this paper, we presented and implemented for the first time a new approach termed the multi-domain Legendre–Gauss–Lobbatto Bivariate spectral quasilinearization method for solving general nonlinear evolution smooth partial differential equations. The method independently used Lagrange interpolation polynomials based on Legendre–Gauss–Lobatto grid points in both space and time. The time domain was further divided into small sub-intervals and the linearized evolution equations were solved in each sub-interval.

Numerical simulations were carried out on the Burgers–Fisher equation, Fitzhugh–Nagumo equation, modified KdV–Burger equation, highly nonlinear modified KdV equation, the Fisher equation and Burgers–Huxley equation. The MD-LGL-BSQLM method converged to a more accurate solution than the LGL-BSQLM method over large time domains. The MD-LGL-BSQLM method also used minimal computational time compared to the LGL-BSQLM method. We can therefore conclude that the MD-LGL-BSQLM method is more accurate and uses less computational time compared to the LGL-BSQLM method.

This study contributes to new techniques that can be used for solving NPDE's over large time domains. Further studies remain to establish whether the method can be used successfully to find solutions of nonlinear coupled systems of differential equations, including hyperbolic equations.

Acknowledgment

This work was supported by the University of KwaZulu-Natal. The authors would like to thank the reviewers for the constructive comments which has improved the quality of the paper.

References

Abbasbandy, S. [2008] "Soliton solutions for the Fitzhugh–Nagumo equation with the homotopy analysis method," *Appl. Math. Model.* **32**(12), 2706–2714.

Adibi, H. and Rismani, A. M. [2010] "On using a modified Lagrange-spectral method for solving singular IVPs of Lane-Emden type," *Comput. Math. Appl.* **60**, 2126–2130.

Batiha, B., Noorani, M. S. M. and Hashim, I. [2008] "Application of variational iteration method to the generalized Burgers–Huxley equation," *Chaos Solitons Fractals* **36**(3), 660–663.

Bellman, R. E. and Kalaba, R. E. [1965] *Quasilinearization and Nonlinear Boundary-Value Problems* (Elsevier, New York, USA).

Bhrawy, A. H., Zaky, M. A. and Baleanu, D. [2015] "New numerical approximations for space-time fractional Burgers equations via a Legendre spectral-collocation method," *Rom. Rep. Phys.* **67**(2), 1–13.

Burgers, J. M. [1948] "A mathematical model illustrating the theory of turbulence," *Adv. Appl. Mech. I* 171–199.

Canuto, C., Hussaini, M. Y., Quarteroni, A. and T. A. Zang, T. A. [1988] *Spectral Methods in Fluid Dynamics* (Springer-Verlag, Berlin).

Canuto, C., Hussaini, M. Y., Quarteroni, A. and Zang, T. A. [2007] *Spectral Methods, Evolution to Complex Geometries and Applications to Fluid Dynamics* (Springer-Verlag, Berlin).

Cole, J. D. [1951] "On a quasilinear parabolic equations occurring in aerodynamics," *Quart. Appl. Math.* **9**, 225–236.

Darvishi, M. T., Khani, F. and Kheybari, S. [2007] "A numerical solution of the KdV-Burgers equation by spectral collocation method and Darvishis Preconditionings," *Int. J. Contemp. Math. Sci.* **2**(22), 1085–1095.

Darvishi, M. T., Kheybari, S. and Khani, F. [2008] "Spectral collocation method and Darvishis preconditionings to solve the generalized Burgers–Huxley equation," *Communic. Nonlinear Sci. Numer. Simul.* **13**(10), 2091–2103.

Dehghan, M., Heris, J. M. and Saadatmandi, A. [2010] "Application of semi-analytic methods for the Fitzhugh–Nagumo equation, which models the transmission of nerve impulses," *Math. Methods Appl. Sci.* **33**(11), 1384–1398.

Dehghan, M., Saray, B. N. and Lakestani, M. [2012] "Three methods based on the interpolation scaling functions and the mixed collocation finite difference schemes for the numerical solution of the nonlinear generalized Burgers–Huxley equation," *Math. Comput. Model.* **55**(3), 1129–1142.

Dlamini, P. G., Khumalo, M. and Motsa, S. S. [2013] "A Note on the Multi-stage Spectral Relaxation Method for Chaos Control and Synchronization," *Int. J. Nonlinear Sci. Numer. Simul.* **15**, 289–298.

Elnagar, G. N. and Kazemi, M. A. [1997] "Pseudospectral Lagrange-based optimal computation of nonlinear constrained variational problems," *J. Comput. Appl. Math.* **88**, 363–375.

Fahroo, F. and Ross, M. [2001] "Costate Estimation by a Lagrange Pseudospectral Method," *J. Guid. Control Dyn.* **24**, 270–277.

Fan, E. G. [2002] "Travelling wave solutions for nonlinear equations using symbolic computation," *Comput. Math. Appl.* **43**, 671–680.

Golbabai, A. and Javidi, M. [2005] "A spectral domain decomposition approach for the generalized Burger's–Fisher equation," *Chaos Solitons Fractals* **39**, 385–392.

Golbabai, A. and Javidi, M. [2009] "A spectral domain decomposition approach for the generalized Burger's–Fisher equation," *Chaos Solitons Fractals* **39**(1), 385–392.

Hariharan, G., Kannan, K., and Sharma, K. R. [2009] "Haar wavelet method for solving Fishers equation," *Appl. Math. Comput.* **211**(2), 284–292.

Hashim, I., Noorani, M. S. M. and Al-Hadidi, M. R. S. [2006] "Solving the generalized Burgers–Huxley equation using the Adomian decomposition method," *Math. Comput. Model.* **43**, 1404–1411.

Helal, M. A. and Mehanna, M. S. [2006] "A Comparison between two different methods for solving KdV-Burgers equation," *Chaos Solitons Fractals* **28**, 320–326.

Javidi, M. [2006] "Spectral collocation method for the solution of the generalized Burger-Fisher equation," *Appl. Math. Comput.* **174**(1), 345–352.

Jiwari, R., Pandit, S. and Mittal, R. C. [2012] "Numerical simulation of two-dimensional sine-Gordon solitons by differential quadrature method," *Comput. Phys. Commun.* **183**, 600–616.

Kheiri, H. and Ebadi, D. G. [2010] "Application of the (G'/G)-expansion method for the Burgers, Fisher, and Burgers–Fisher equations," *Acta Univ. Apulensis* **24**, 35–44.

Kurulay, M. and Bayram, M. [2010] "Approximate analytical solution for the fractional modified KdV by differential transform method," *Commun. Nonlinear Sci. Numer. Simul.* **15**(7), 1777–1782.

Li, H. and Guo, Y. [2006] "New exact solutions to the Fitzhugh–Nagumo equation," *Appl. Math. Comput.* **180**, 524–558.

Li, J., Ma, H. and Sun, W. [2000] "Error analysis for solving the Korteweg-de Vries equation by a Lagrange pseudo-spectral method," *Numer. Methods Partial Differ. Eq.* **16**, 513–534.

Ma, H. [2001] "Composite Lagrange-Laguerre pseudospectral approximation in unbounded domains," *IMA J. Numer. Anal.* **21**, 587–602.

Ma, H. and Sun, W. [2000] "A Lagrange-Petrov-Garlekin and Chebyshev collocation method for third order differential equations," *SIAM J. Numer. Anal.* **38**, 1425–1438.

Mickens, R. E. [1997] "Relation between the time and space step-sizes in nonstandard finite-difference schemes for the Fisher equation," *Numer. Methods Partial Differ. Equ.* **13**(1), 51–55.

Mittal R. C. and Jiwari, R. [2009a] "Study of Burger-Huxley equation by differential quadrature method," *Int. J. Appl. Math. Mech.* **5**(8), 1–9.

Mittal R. C. and Jiwari R. [2009b] "Differential quadrature method for two dimensional Burgers' equations," *Int. J. Comput. Methods Eng. Sci. Mech.* **10**, 450–459.

Mittal, R. C. and Tripathi, A. [2014] "Numerical solutions of generalized Burgers–Fisher and generalized Burgers–Huxley equations using collocation of cubic B-splines," *Int. J. Comput. Math.*, http://dx.doi.org/10.1080/00207160.2014.920834.

Moghimi, M. and Hejazi, F. S. [2007] "Variational iteration method for solving generalized BurgerFisher and Burger equations," *Chaos Solitons Fractals* **33**(5), 1756–1761.

Motsa, S. S. [2012] "A new piecewise-quasilinearization method for solving chaotic systems of initial value problems," *Cent. Eur. J. Phys.* **10**, 936–946.

Motsa, S. S., Dlamini, P. and Khumalo, M. [2012] "Solving hyper-chaotic systems using the spectral relaxation method," *Abs. Appl. Anal.* **2012**, Article ID 203461.

Motsa, S. S., Dlamini, P. and Khumalo, M. [2013] "A new multi-stage spectral relaxation method for solving chaotic initial value systems," *Nonlinear Dyn.* **72**, 265–283.

Motsa, S. S., Magagula, V. M. and Sibanda, P. [2014] "A bivariate Chebyshev spectral collocation quasilinearization method for nonlinear evolution parabolic equations," *Sci. World J.* **2014**, Article ID 581987, 13.

Motsa, S. S. and Sibanda, P. [2012] "A multistage linearisation approach to a four-dimensional hyper-chaotic system with cubic nonlinearity," *Nonlinear Dyn.* **70**, 651–657.

Nawaz R, Ullah H, Islam, S. and Idrees, M. [2013] "Application of optimal homotopy asymptotic method to Burger equations," *J. Appl. Math.*, http://dx.doi.org/10.1155/2013/387478, doi: 10.1155/2013/93515424415902.

Olmos, D. and Shizgal, B. D. [2006] "A pseudospectral method of solution of Fisher's equation," *J. Comput. Appl. Math.* **193**(1), 219–242.

Pandey, R. K, Kumar, N., Bhardwaj, A. and Dutta, G. [2012] "Solution of Lane-Emden type equations using Lagrange operational matrix of differentiation," *Appl. Math. Comput.* **218**, 7629–7637.

Polyanin, A. D. and Zaitsev, V. F. [2004] *Handbook of Nonlinear Partial Differential Equations* (CRC Press, US).

Rismani, A. M. and Monfared, H. [2012] "Numerical solution of singular IVPs of Lane-Emden type using a modified Lagrange-spectral method," *Appl. Math. Model.* **36**, 4830–4836.

Shamsi, M. and Dehghan, M. [2006] "Recovering a time-dependent coefficient in a parabolic equation from overspecified boundary data using the pseudospectral Lagrange method, *Wiley Periodicals*, 196–209.

Shamsi, M. and Dehghan, M. [2012] "Determination of a control function in three-dimensional parabolic equations by Legendre pseudospectral method," *Numer. Methods Partial Differ. Equ.* **28**(1), 74–93.

Shateyi, S., Motsa, S. S. and Khan, Y. [2014] "A new piecewise spectral homotopy analysis of, the Michaelis-Menten enzymatic reactions model," *Numer. Algorithms* **66**, 495–510.

Sherratt, J. [1998] "On the transition from initial data traveling waves in the Fisher-KPP equation," *Dyn. Stab. Syst.* **13**, 167–174.

Tian B. and Zong, Q. [2010] "Optimal guidance for reentry vehicles based on indirect Lagrange pseudospectral method," *Acta Astronautica* **68**, 1176–1184.

Van Gorder, R. A. [2012] "Gaussian waves in the Fitzhugh–Nagumo equation demonstrate one role of the auxiliary function $H(x,t)$ in the homotopy analysis method," *Commun. Nonlinear Sci. Numer. Simul.* **17**(3), 1233–1240.

Van Gorder, R. A., and Vajravelu, K. [2010] "A variational formulation of the Nagumo reaction-diffusion equation and the Nagumo telegraph equation," *Nonlinear Anal.: Real World Appl.* **11**(4), 2957–2962.

Voigt, R. G., Gottlieb, D. and Hussaini, M. Y. [1984] *Spectral Methods for Partial Differential equations* (SIAM-CBMS, Philadephia).

Wang, T. J. and Guo, B. Y. [2008] "Composite generalized Laguerre-Lagrange pseudospectral method for Fokker-Planck equation in an infinite channel," *Appl. Numer. Math.* **58**, 1448–1466.

Wang, X. Y., Zhu, Z. S. and Lu, Y. K. [1990] "Solitary wave solutions of the generalised Burgers–Huxley equation," *J. Phys. A, Math.* **23**, 271–274.

Wazwaz, A. M. and Gorguis, A. [2004] "An analytic study of Fisher's equation by using Adomian decomposition method," *Appl. Math. Comput.* **154**, 609–620.

Wu, H., Ma, H. and Li, H. [2003] "Optimal error estimates of the Chebyshev-Lagrange spectreal method for solving generalized Burgers equation," *SIAM J. Numer. Anal.* **41**, 659–672.

Wu, B. and Wang, D. [2009] "Nonlinear optimization of low-thrust trajectory for satellite formation: Lagrange pseudospectral approach," *J. Guid. Control Dyn.* **32**, 1371–1381.